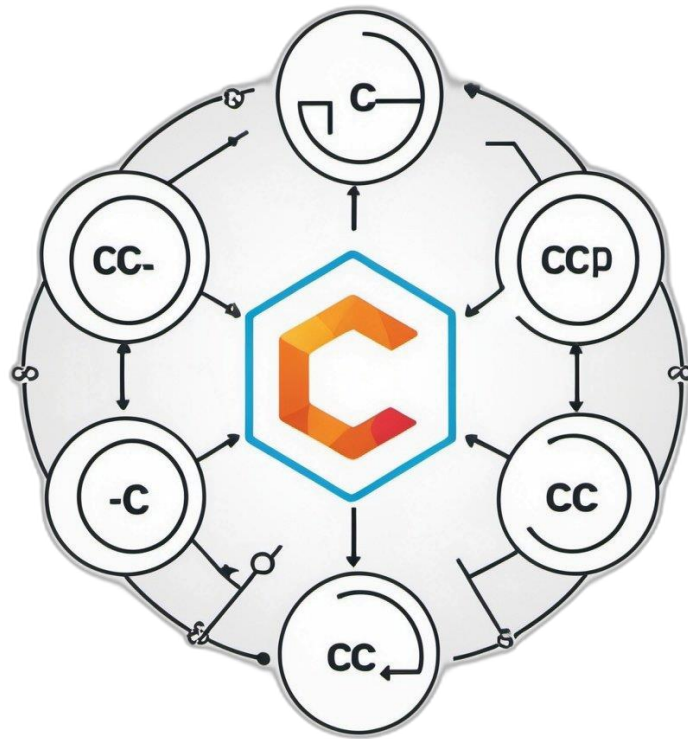


ՀԱՅ-ՌՈՒՄԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ
ՄԱԹԵՄԱՏԻԿԱՅԻ ԵՎ ԻՆՖՈՐՄԱՏԻԿԱՅԻ
ԻՆՍՏԻՏՈՒՏ

Սևակ Սարգսյան Հայկ Ասլանյան



«Ալգորիթմներ և ալգորիթմական լեզուներ»

Գործնական պարասպմունքների ուղեցույց

Մաս I





**РОССИЙСКО-АРМЯНСКИЙ (СЛАВЯНСКИЙ) УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАТИКИ**

СЕВАК САРГСЯН, АЙК АСЛАНЯН

АЛГОРИТМЫ И АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ
Руководство для практических занятий
Часть I

Ереван–2024

Издательство РАУ

**ՀԱՅ-ՌՈՒՄԱԿԱՆ (ՍԼԱՎՈՆԱԿԱՆ) ՀԱՄԱԼՍԱՐԱՆ
ՄԱԹԵՄԱՏԻԿԱՅԻ ԵՎ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԻՆՍՏԻՏՈՒՏ**

ՍԵՎԱԿ ՍԱՐԳՍՅԱՆ, ՀԱՅԿ ԱՍԼԱՆՅԱՆ

**ԱԼԳՈՐԻԹՄՆԵՐ և ԱԼԳՈՐԻԹՄԱԿԱՆ ԼԵԶՈՒՆԵՐ»
Գործնական պարապմունքների ուղեցույց
Մաս I**

Երևան–2024

ՀՌՀ հրատարակչություն

ՀՏԴ 004(07)
ԳՄԴ 32.97g7
Ս 259

*Հրատարակման երաշխավորվում Հայ-
Ռուսական համալսարանի
Գիտատեխնիկական և Խմբագրա-
Հրատարակչական խորհրդի կողմից*

Պատասխանատու խմբագիր՝ տ.գ.թ. Ջ.Ա.Հակոբյան
Գրախոս՝ Երևանի պետական համալսարանի
Ինֆորմատիկայի և կիրառական մաթեմատիկայի ֆակուլտետի
Ծրագրավորման և ինֆորմացիոն տեխնոլոգիաների ամբիոնի դասախոս, տեխ.գիտ.թեկնածու, դոցենտ
Ա.Ս.Հովակիմյան

Սարգսյան Ս., Ասլանյան Հ.
Ս 259 Ալգորիթմներ և ալգորիթմական լեզուներ: Գործնական պարապմունքների ուղեցույց Մաս 1. /
Ս. Սարգսյան, Հ. Ասլանյան. – Եր.: ՀՌՀ, 2024. – 75 էջ:

Սույն աշխատանքի նպատակն է ուղղորդել «Ալգորիթմներ և ալգորիթմական լեզուներ» առարկայի գործնական աշխատանքները: Յուրաքանչյուր դասի համար բերվում է համառոտ նկարագիր ուսումնասիրվող թեմայի վերաբերյալ և կողի որոշ օրինակներ: Այնուհետ ներկայացվում են մի շարք խնդիրներ, որոնց մի մասը պետք է լուծվեն գործնական աշխատանքի ժամանակ, իսկ մնացած մասը հանձնարարվեն որպես տնային աշխատանք:

ՀՏԴ 004(07)
ԳՄԴ 32.97g7

© ՀՌՀ հրատարակչություն, 2024

ISBN 978-9939-67-340-0

© Սարգսյան Ս.Ս., Ասլանյան Հ.Կ., 2024

Նախաբան

Այս ձեռնարկում ներկայացված են Հայ-Ռուսական համալսարանի, Մաթեմատիկայի և Ինֆորմատիկայի ինստիտուտում դասավանդվող «Ալգորիթմներ և ալգորիթմական լեզուներ» առարկայի շրջանակում լուծվող գործնական խնդիրները: Այս առարկայի նպատակն է ուսանողներին սովորեցնել ալգորիթմական մտածողություն: Նպատակին հասնելու համար ուսանողները սովորում են C և C++ ծրագրավորման լեզուները՝ լուծելով տարբեր ալգորիթմական խնդիրներ: C լեզուն պարզ է և ապահովում է համակարգչի ռեսուրսները կառավարելու հնարավորություն, ինչը թույլ է տալիս հասկանալ համակարգչի աշխատանքի հիմնական սկզբունքները: C++ լեզուն, որը ուսումնասիրվում է առարկայի երկրորդ մասում, ավելի շատ ֆունկցիոնալ հնարավորություններ է տրամադրում և թույլ է տալիս հասկանալ օբյեկտակողմնորոշված ծրագրավորման հիմունքները:

Դաս 1. Ծանոթություն C լեզվի հետ

Ներկայացվում է C լեզուն, նախնական կողի թարգմանության, ինչպես նաև ստացված ծրագրի կատարման պրոցեսը: Բերվում է առաջին ծրագրի օրինակը, որը ներկայացված է ներքևում: Ներկայացվում է կցված stdio գրադարանը, էլքագրման համար նախատեսված printf ֆունկցիան, ինչպես նաև հատուկ սիմվոլները՝ '\n', '\t'.

```
// The first program "Hello World"
#include <stdio.h>
int main(void)
{
    printf("Hello World\n");
}
```

Ներկայացվում է ամբողջ թվերի համար նախատեսված int տիպը: Ներկայացվում են որոշ պարզ գործողություններ ամբողջ տիպերի հետ: Պատմվում է փոփոխականի տեսանկյունից տիրույթի մասին: Ներքևում բերված է կողի օրինակ, որը թույլ է տալիս արտածել ամբողջ տիպի փոփոխականի թվային արժեքը:

```
#include <stdio.h>
int main(void)
{
    int x = 10;
    printf("The value of x is %d\n", x);
}
```

Խնդիր 1

Արտածեք ձեր անունը և ազգանունը առանձին տողերում:

Խնդիր 2

Արտածել հետևյալ պատկերները.

```
 * * * *      *      *
 *   *      * * *      * *
 *   *      * * * * *      * * *
 * * * *      * * *      * * * *
                *      * * * * *
```

Խնդիր 3

Հայտարարել int տիպի երկու ամբողջ թվեր և հաշվել դրանց գումարը, տարբերությունը և արտադրյալ: Արտածել ստացված արդյունքները: Ներքևում բերվում է գումարը հաշվող կոդի օրինակը:

```
#include <stdio.h>
int main(void)
{
    int x = 10;
    int y = 20;
    int sum = x + y;
    printf("%d\n", sum);
}
```

Խնդիր 4

Հաշվել տրված ամբողջ թվի քառակուսին և խորանարդը: Արտածել ստացվածը արդյունքները:

Խնդիր 5

Հաշվել $ax^2 + bx + c$ արտահայտության արժեքը, եթե տրված են a, b, c գործակիցները և x փոփոխականը: Հայտնի է, որ a, b, c, x թվերը ամբողջ թվեր են:

Դաս 2. Տիպեր, մոդիֆիկատորներ, գործողություններ, սահմաններ, տվյալների մուտքագրում և ելքագրում

Ամբողջ տիպերի մոդիֆիկատորներ, հիմնականգործողություններ

Ներկայացվում են ամբողջ տիպի մոդիֆիկատորները և հիմնական գործողությունները: Բերվում են տարբեր մոդիֆիկատորներով ամբողջ տիպերի արտաձման միջոցները: Ներքևում բերվում են մոդիֆիկատորների օգտագործման և փոփոխականների արտաձման օրինակներ:

```
#include <stdio.h>
int main ()
{
    short s = 10;
    int i = 20;
    long l = 30;
    long long ll = 40;

    printf("short: %d\n", s);
    printf("int: %d\n", i);
    printf("long: %ld\n", l);
    printf("long long: %lld\n", ll);

    unsigned short us = 100;
    unsigned int ui = 200;
    unsigned long ul = 300;
    unsigned long long ull = 400;

    printf("unsigned short: %u\n", us);
    printf("unsigned int: %u\n", ui);
    printf("unsigned long: %lu\n", ul);
    printf("unsigned long long: %llu\n", ull);
}
```

```

#include <stdio.h>
int main(void)
{
    int a = 10, b = 20, d = 15;
    int sum = a + b;
    int diff = a - b;
    int mul = a * d;
    int div = d / a;
    int div_rem = d % a;
    printf("Sum = %d\n", sum);
    printf("Diff = %d\n", diff);
    printf("Mul = %d\n", mul);
    printf("Div = %d\n", div);
    printf("Div. rem. = %d\n", div_rem);
}

```

Ամբողջ տիպի փոփոխականների սահմաններ

```

#include <stdio.h>
#include <limits.h>
int main () {
    printf("The minimum value of SHORT INT = %hd\n", SHRT_MIN);
    printf("The maximum value of SHORT INT = %hd\n", SHRT_MAX);

    printf("The minimum value of INT = %d\n", INT_MIN);
    printf("The maximum value of INT = %d\n", INT_MAX);

    printf("The minimum value of LONG INT = %ld\n", LONG_MIN);
    printf("The maximum value of LONG INT = %ld\n", LONG_MAX);

    printf("The minimum value of LONG LONG INT = %lld\n", LLONG_MIN);
    printf("The maximum value of LONG LONG INT = %lld\n", LLONG_MAX);
}

```

Առանց նշանի թվեր

```
#include <stdio.h>
#include <limits.h>

int main () {
    printf("The maximum value of UNSIGNED SHORT INT = %u\n",
    USHRT_MAX);
    printf("The maximum value of UNSIGNED INT = %u\n", UINT_MAX);
    printf("The maximum value of UNSIGNED LONG INT = %lu\n",
    ULONG_MAX);
    printf("The maximum value of UNSIGNED LONG LONG INT = %llu\n",
    ULLONG_MAX);
}
```

Տվյալների մուտքագրում

Ներկայացվում են տվյալների մուտքագրման միջոցները, որոնցից մեկ օրինակ բերված է ներքևում:

```
#include <stdio.h>
int main ()
{
    int x;
    printf("Enter an integer ");
    scanf("%d", &x);
    printf("Displaying number %d", x);
}
```

Սահող կետով թվեր

Ներկայացվում են իրական թվերի՝ համակարգչում պահպանման մոտեցումները, որոնց այսուհետ կհղվենք սահող կետով թվեր: Ներկայացվում են float և double տիպերը: Բերվում են սահող կետերով թվերի հետ կատարվող հիմնական գործողությունները և դրանց արտաձման միջոցները:

```

#include <stdio.h>
int main(void)
{
    float a = 1.2, b = 0.3;
    float sum = a + b;
    float diff = a - b;
    float mul = a * b;
    float div = a / b;
    printf("sum = %f\n", sum);
    printf("diff = %f\n", diff);
    printf("mul = %f\n", mul);
    printf("div = %f\n", div);
}

```

Ներկայացվում են նաև սահող կետով թվերի պահպանման ժամանակ առաջացող ճշտության կորուստները:

```

#include <stdio.h>

int main(void)
{
    float f = 3.999999;
    f = f * f;
    double d = 3.999999;
    d = d * d;
    long double ld = 3.999999;
    ld = ld * ld;
    printf("f = %.20f d = %.20f, ld = %.20LF", f, d, ld);
}

```

Խնդիր 1

Մուտքագրել երկու ամբողջ թիվ և արտածել նրանց գումարը, տարբերությունն ու արտադրյալը:

Խնդիր 2

Մուտքագրել երկու/երեք բնական թիվ և արտածել նրանց միջին թվաբանականը և միջին երկրաչափականը:

Խնդիր 3

Մուտքագրել քառանիշ թիվ: Արտածել թվի թվանշանները՝ դասավորված հակառակ կարգով (օրինակ՝ 1265, 5621):

Խնդիր 4

Մուտքագրել քառանիշ թիվ: Արտածել թվի թվանշանները, դրանց միջև տեղադրելով գծիկներ (օրինակ՝ 6508, 6-5-0-8):

Խնդիր 5

Մուտքագրել օրերի քանակ և արտածել այդ օրերը վերածած տարիների, շաբաթների:

Խնդիր 6

Մուտքագրել եռանիշ ամբողջ թիվ և արտածել առաջին, երկրորդ և երրորդ թվանշանները:

Խնդիր 7

Մուտքագրել եռանիշ ամբողջ թիվ և արտածել բոլոր թվանշանների գումարը:

Խնդիր 8

Մուտքագրել օրվա ժամը՝ օրվա սկզբից անցած վարկյանների տեսքով: Արտածել թե քանի լրիվ ժամ և րոպե է անցել այդ պահին:

Խնդիր 9

Արտածել համապատասխան ժամսլաքների անկյունները (աստիճաններով) օրվա սկզբում դրանց դիրքերի և $h(\text{ժամ}):m(\text{րոպե}):s(\text{վայրկյան})$ ժամին համապատասխան դիրքերի միջև:

$h(0 \leq h < 12)$, $m(0 \leq m < 60)$ և $s(0 \leq s < 60)$ արժեքները մուտքագրվում են:

Խնդիր 10

Մուտքագրել խորանարդի կողմը սահող թվով: Արտածել խորանարդի ծավալը և կողմնային մակերևույթի մակերեսը:

Խնդիր 11

Մուտքագրել շրջանագծի տրամագիծը և արտածել նրա մակերեսը 5 նիշի ճշտությամբ:

Խնդիր 12

Մուտքագրել եռանկյան կողմերը և արտածել եռանկյան պարագիծն ու մակերեսը:

Խնդիր 13

Մուտքագրել հարթության վրա երկու կետերի կոորդինատները և արտածել դրանց միջև հեռավորությունը:

Դաս 3. Սիմվոլային տիպ, տվյալների կոդավորում, sizeof օպերատոր

Սիմվոլների կոդավորում

Ներկայացվում է սիմվոլների կոդավորման (ASCII կոդ) հիմնական սկզբունքները և char սիմվոլային տիպը: Ներքևում բերվում է char տիպի օգտագործման օրինակ:

```
#include <stdio.h>
int main(void)
{
    char chr;
    printf("Enter a character: ");
    scanf("%c", &chr);

    // When %c is used, a character is displayed
    printf("You entered %c.\n",chr);

    // When %d is used, ASCII code value is displayed
    printf("ASCII value is %d.", chr);
}
```

Սիմվոլային տիպի հետ թույլատրվում են կատարել թվաբանական գործողություններ՝ ինչպես ամբողջ տիպով փոփոխականների հետ: Ներքևում բերվում է օրինակ:

```
#include <stdio.h>

int main(void)
{
    char x = 'a';
    char y = x + 8;
    char z = y - 3;
    printf("%c : %c", y, z);
}
```

Հաջորդ օրինակը արդեն ցույց է տալիս մուտքագրված լատիներեն մեծատառի համար համապատասխան փոքրատառի ստացումը:

```
#include <stdio.h>

int main(void)
{
    char upperCase;
    printf("Enter an uppercase character ");
    scanf("%c", &upperCase);
    char lowerCase = upperCase + 32; // upperCase + 'a' - 'A';
    printf("The lowercase character is %c\n", lowerCase);
}
```

sizeof օպերատոր

Ներկայացվում է *sizeof* օպերատորը՝ տիպերի չափերը ստանալու համար: Ներքևում բերվում է օգտագործման օրինակ:

```
#include <stdio.h>

int main(void)
{
    printf("%lu\n", sizeof(char));
    printf("%lu\n", sizeof(int));
    printf("%lu\n", sizeof(float));
    printf("%lu\n", sizeof(double));

    int a = 0;
    double d = 10.21;
    printf("%d\n%lu", a+(int)d, sizeof(a + (int)d));
}
```

Տվյալների կոդավորման սկզբունքներ

Ներկայացվում է համակարգչում տվյալների կոդավորման հիմնական մոտեցումները՝ ուղիղ, հակադարձ և լրացուցիչ կոդեր: Ներքևում բերվում է

օրինակներ տվյալների տասնվեցական, տասական, ութական և երկուականներկայացումների համար:

Տասական	Երկուական	Ութական	Տասնվեցական
0	00000000	00	0x0
1	00000001	011	0x1
8	00001000	015	0xA
12	00001100	023	0x10
36	00100100	045	0x22
17	00010001	0100	0x99
254	11111110	0156	0xFF

Ներքևում ևս մի քանի օրինակներ բերվում են արդեն տվյալների երկուականներկայացման ուղիղ, հակադարձ և լրացուցիչ կոդերի համար:

Տասական	Ուղիղ	Հակադարձ	Լրացուցիչ
8	00001000	11110111	11111000
9	00001001	11110110	11110111
-9	10001001	11110110	11110111
-12	10001100	11110011	11110100

Ներկայացվում է, որ առանց նշանով թվերի դեպքում չի օգտագործվում նշանի բիթը, ինչը թույլ է տալիս պահպանել ավելի մեծ դրական թվեր:

```
#include <stdio.h>
int main(void)
```

```
{
    int a = 4294967295; // wrong
    unsigned int b = 4294967295; // ok
    printf("a = %d, b = %u", a, b);
}
```

Խնդիր 1

Մուտքագրված լատիներեն փոքրատառի համար արտածել համապատասխան մեծատառը:

Խնդիր 2

Մուտքագրված ASCII կոդի համար արտածել համապատասխան սիմվոլը:

Խնդիր 3

Մուտքագրված լատիներեն մեծատառի համար արտածել համապատասխան փոքրատառը:

Խնդիր 4

Մուտքագրված սիմվոլի համար արտածել համապատասխան ASCII կոդը:

Խնդիր 5

Մուտքում ստացված *int*, *float*, *double*, *char* տիպի փոփոխականների համար արտածել դրանց չափերը, այնուհետև կատարելով թվաբանական գործողություններ դրանց հետ՝ արտածել ստացված նոր արժեքների տիպերիչափերը, օգտագործելով *sizeof* օպերատորը:

Խնդիր 6

Աղյուսակում նշված երկուական ներկայացման համար ստանալ տասական ներկայացումը: Լուծել խնդիրը երկու դեպքերի համար, երբ տվյալները տրամադրված են ուղիղ կոդով և լրացուցիչ կոդով:

Երկուական կոդ	Լրացուցիչ կոդով դեպք	Ուղիղ կոդով դեպք
0000 0000	0	0
0000 0001	1	1
1111 1111	-1	-127
0111 0111	119	119
1000 1001	-119	-9
1000 0001	-127	-1

Դաս 4. Բիթային, համեմատման և ճյուղավորման օպերատորներ

Բիթային օպերատորներ

Ներկայացվում են հիմնական բիթային օպերատորները և դրանց հետաշխատելու օրինակներ:

```
#include <stdio.h>

int main(void)
{
    int a = -10, b = 21;
    printf("Output = %d\n", a&b); // bitwise and

    printf("Output = %d\n", a|b); // bitwise or
    printf("Output = %d\n", a^b); // bitwise xor
    printf("Output = %d\n", ~a); // bitwise complement
    int i = 2;
    printf("Right shift by %d: %d\n", i, a>>i);
    printf("Left shift by %d: %d\n", i, b<<i);
}
```

Ճյուղավորման պարզագույն օպերատոր

Ներկայացվում է ծրագիր ճյուղավորման պարզագույն if/else օպերատոր, որի կիրառման օրինակներից երկուսը բերվում է հաջորդիվ:

```
// Program to display a number if it is negative
#include <stdio.h>
int main(void)
{
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);
}
```

```

// true if number is less than 0
if (number < 0)
{
    printf("You entered %d.\n", number);
}
}

```

```

// Check whether an integer is odd or even
#include <stdio.h>

int main(void)
{
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);
    // True if the remainder is 0
    if (number % 2 == 0)
    {
        printf("%d is an even integer.", number);
    }
    else
    {
        printf("%d is an odd integer.", number);
    }
}

```

Համեմատման օպերատորներ

Ներկայացվում են համեմատման օպերատորները (<, >, ==, !=, >=, <=) և դրանց հետ աշխատելու օրինակներ, որոնցից մեկը բերված է ներքևում:

```

// Working of relational operators
#include <stdio.h>
int main(void)
{
    int a = 5, b = 5, c = 10;

```

```

printf("%d == %d is %d \n", a, b, a == b);
printf("%d == %d is %d \n", a, c, a == c);
printf("%d > %d is %d \n", a, b, a > b);
printf("%d > %d is %d \n", a, c, a > c);
printf("%d < %d is %d \n", a, b, a < b);
printf("%d < %d is %d \n", a, c, a < c);
printf("%d != %d is %d \n", a, b, a != b);
printf("%d != %d is %d \n", a, c, a != c);
printf("%d >= %d is %d \n", a, b, a >= b);
printf("%d >= %d is %d \n", a, c, a >= c);
printf("%d <= %d is %d \n", a, b, a <= b);
printf("%d <= %d is %d \n", a, c, a <= c);
}

```

Համեմատման օպերատորների
օպերատորներում

օգտագործումը

ճյուղավորման

Ներկայացվում են ինչպես կարող են օգտագործվել համեմատման օպերատորները ճյուղավորման օպերատորներում: Ստորև բերված օրինակում օգտագործվում է նաև else if.

```

#include <stdio.h>

int main(void)
{
    int x; scanf("%d",
    &x);if (x > 0) {
        printf("x is positive\n");
    } else if (x == 0) { printf("x
    is zero\n");
    } else {
        printf("x is negative\n");
    }
}

```

Խնդիր 1

Ներմուծել *int x, y;* *float z;* *char c;* փոփոխականների արժեքները: Արտածել հետևյալ արտահայտությունների արժեքները:

1. $x \gg y$
2. $x \& y / \sim x$
3. $x / \sim x$
4. $x \& x + y$
5. $c = '1'; c \ll 2 \gg 2$
6. $++x - y \geq 5 \&\& y$

Խնդիր 2

Մուտքագրված ամբողջ թիվը բաժանել/բազմապատկել 2-ի/ով, 4-ի/ով, 8-ի/ով՝ օգտագործելով միայն բիթային տեղաշարժի գործողությունները:

Խնդիր 3

Մուտքագրված ամբողջ թիվը բազմապատկել 6-ով միայն օգտագործելով գումարում և բիթային տեղաշարժ գործողությունները:

Խնդիր 4

Մուտքագրված ամբողջ թվի առաջին 5 բիթերը լրացնել զրոներով:

Խնդիր 5

Մուտքագրված ամբողջ թվի 7-րդ բիթի արժեքը սարքել 1:

Խնդիր 6

Մուտքագրված թվի երկուական ներկայացման մեջ գտեք ամենաաջիկոքը, որը պարունակում է 1 արժեքը:

Խնդիր 7

Փոխանակել մուտքագրված երկու փոփոխականների արժեքները առանց երրորդ փոփոխական օգտագործելու:

Խնդիր 8

Մուտքագրել a և b բնական թվերը, արտածել "yes", եթե երկուսն էլ զույգ են:

Խնդիր 9

Մուտքագրել a և b բնական թվերը, արտածել "yes", եթե դրանցից գոնեմեկը կենտ թիվ է:

Խնդիր 10

Մուտքագրել $input_1$, $input_2$ և $input_3$ ամբողջ թվերը: Արտածել "*condition is correct*", եթե

1. երեքն էլ դրական են,
2. նրանցից գոնե մեկը դրական է,
3. նրանցից ճիշտ մեկը դրական է,
4. նրանցից ճիշտ երկուսը դրական են,
5. նրանցից գոնե երկուսն իրար հավասար են,

Հակառակ դեպքում "*condition isn't correct*":

Դաս 5. Ներդրված ճյուղավորման օպերատորներ, switch/case օպերատոր, enum տիպ

Ներդրված ճյուղավորման օպերատորներ

Ներկայացվում են ներդրված ճյուղավորման օպերատորների կառուցվածքը: Ներքևում բերվում է օրինակ:

```
// Program example of nested if else
#include <stdio.h>
int main(void)
{
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);
    if (number1 >= number2)
    {
        if (number1 == number2)
        {
            printf("Result: %d = %d", number1, number2);
        }
        else
        {
            printf("Result: %d > %d", number1, number2);
        }
    }
    else
    {
        printf("Result: %d < %d", number1, number2);
    }
}
```

switch/case օպերատոր

Ներկայացվում է *switch/case* օպերատորը և դրա հիմնական կիրառությունները, որի օրինակներից մեկը ներկայացված է հաջորդիվ:

```
// Program to create a simple calculator
#include <stdio.h>

int main(void)
{
    char op;
    double n1, n2;
    printf("Enter an op (+, -, *, /): ");
    scanf("%c", &op);

    scanf("%lf %lf", &n1, &n2);
    switch(op)

        case '+':
            printf("%.1lf %.1lf %.1lf", n1, n2, n1 + n2);
            break;
        case '-':
            printf("%.1lf %.1lf %.1lf", n1, n2, n1 - n2);
            break;

        // op doesn't match any case constant +, -, *, /
        default:
            printf("Error! op is not correct");
    }
    case '*':
        printf("%.1lf %.1lf %.1lf", n1, n2, n1 * n2);
        break;
    case '/':
        printf("%.1lf %.1lf %.1lf", n1, n2, n1 / n2);
        break;
```

enum տիպ

Ներկայացվում է enum տիպը և դրա հիմնական կիրառությունները:
Հաջորդիվ բերվում է կիրառության օրինակ:

```
#include <stdio.h>

enum week { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday };

int main(void)
{
    // creating today variable of enum week type
    enum week today;
    today = Wednesday;
    printf("Day %d", today + 1);
}
```

Խնդիր 1

Գրել ծրագիր, որը մոտքում կստանա Էլեկտրաէներգիայի ծախսը և կհաշվարկի վճարման ենթակա գումարը՝ ըստ տրված պայմանների՝

1. Առաջին 50 միավորի համար - 0,50դրամ/միավոր,
2. Հաջորդ 100 միավորների համար - 0,75դրամ/միավոր,
3. Հաջորդ 100 միավորների համար - 1,20դրամ/միավոր,
4. 250 -ից բարձր միավորի համար - 1,50դրամ/միավոր:

Խնդիր 2

Գրել ծրագիր, որը մոտքում կստանա ֆիզիկա, քիմիա, կենսաբանություն, մաթեմատիկա և ինֆորմատիկա առարկաների գնահատականները (0-20 միջակայքում), կհաշվարկի և կարտածի միջինացված առաջադիմության տոկոսը ու կտրամադրի կարգ՝ ըստ տրված պայմանների

1. **Տոկոսը $\geq 90\%$: Ա Կարգ**
2. **Տոկոսը $\geq 80\%$: Բ Կարգ**
3. **Տոկոսը $\geq 70\%$: Գ Կարգ**
4. **Տոկոսը $\geq 60\%$: Դ Կարգ**
5. **Տոկոսը $\geq 40\%$: Ե Կարգ**
6. **Տոկոսը $< 40\%$: Չ Կարգ**

Խնդիր 3

Գրել ծրագիր, որը մոտքում կստանա ամսին համապատասխան թիվը և կարտածի այդ ամսվա օրերի քանակը (օգտագործել *switch/case*):

Խնդիր 4

Գրել ծրագիր, որը մոտքում կստանա ամսին համապատասխան թիվը և կարտածի այդ ամսվա օրերի քանակը (օգտագործել *switch/case*):

Խնդիր 5

Գրել ծրագիր, որը մոտքում կստանա շաբաթվա օրը և կարտածի համապատասխան օրվա անունը (օգտագործել *switch/case*):

Խնդիր 6

Գրել պարզ հաշվիչ, որը մոտքում կստանա 2 իրական թիվ, գործողության տեսակը և կարտածի գործողության արդյունքը(օգտագործել *switch/case*):

Դաս 6. while և do/while ցիկլեր

Ներկայացվում է while ցիկլը, դրա կառուցվածքը և կիրառման տիրույթը: Բերվում է ցիկլի պայմանում համեմատման օպերատորների (<, >, ==, !=, >=, <=) օգտագործման տարբերակները: Ներքևում բերված է պարզցիկլի օրինակ:

```
// Print numbers from 1 to 5
#include <stdio.h>

int main(void)
{
    int i = 1;
    while (i <= 5)
    {
        printf("%d\n", i);
        ++i;
    }
}
```

Հաջորդ օրինակում ցուցադրված է անվերջ աշխատող ցիկլ:

```
// Example of infinite while loop
#include <stdio.h>

int main(void)
{
    int var = 6;
    while (var >=5)
    {
        printf("%d", var);
        var++;
    }
}
```

Բերվում են օրինակներ ցիկլի ներսում ճուղավորման օպերատորներ կիրառելու վերաբերյալ: Մասնավորապես հաջորդիվ բերվում է երկու թվերի ամենամեծ ընդհանուր բաժանարարը գտնող ծրագրի օրինակ:

```

#include <stdio.h>
int main(void)
{
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d",&n1,&n2);
    while(n1 != n2)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
    printf("GCD = %d",n1);
}

```

Ներկայացվում է do/while ցիկլի կառուցվածքը և հիմնական տարբերությունները while ցիկլից: Հաջորդիվ բերվում է պարզ օրինակ:

```

// Program to add numbers until the user enters zero
#include <stdio.h>

int main(void)
{
    unsigned number, sum = 0;
    // the body of the loop is executed at least once
    do
    {
        printf("Enter a number: ");
        scanf("%u", &number);
        sum += number;
    }
    while(number != 0);
    printf("Sum = %u",sum);
}

```

Խնդիր 1

Մուտքագրել K և $N(N > 0)$ ամբողջ թվերը: N անգամ արտածել K թիվը:

Խնդիր 2

Մուտքագրել `n` միանիշ թիվը և արտածել `n`-ին չգերազանցող թվանշանները:

Խնդիր 3

Մուտքագրել `n` բնական թիվը և արտածել `n`-ի՝ 1-ից մինչև 9-րդաստիճանների արժեքները:

Խնդիր 4

Մուտքագրել `n` բնական թիվը և արտածել `n`-ը չգերազանցող բոլորբնական թվերը՝ շրջված/հակառակ կարգով:

Խնդիր 5

Օգտագործելով `while` ցիկլը արտածել լատինատառ այբուբենի փոքրատառերը՝ `a`-ից `z`:

Խնդիր 6

Մուտքագրել `n` բնական թիվը և արտածել `n`-ին չգերազանցող բոլորկենտ/զույգ բնական թվերը:

Խնդիր 7

Մուտքագրել `n` բնական թիվը և արտածել դրա թվանշանների քանակը:

Խնդիր 8

Մուտքագրել `n` բնական թիվը և արտածել `n`-ը չգերազանցող բոլորկենտ/զույգ բնական թվերի գումարը/արտադրյալը:

Խնդիր 9

Գրել ծրագիր, որը արտածում է $y = 2x^2 - 5x - 8$ ֆունկցիայի արժեքների աղյուսակը -4 -ից 4 ինտերվալում: Արգումենտի փոփոխման քայլը 0.5 է:

Խնդիր 10

Մուտքագրել n , l , m բնական թվերը: Ցիկլի մեջ n անգամ մուտքագրել թիվ, և պարզել արդյոք այն պատկանում է (l, m) ինտերվալին:

Խնդիր 11

Մուտքագրել a , b թվերը, և գտնել այդ միջակայքի՝ 7 -ի վրա բաժանվող ամենամեծ և ամենափոքր թվերը:

Խնդիր 12

Մուտքագրել a , b , c բնական թվերը: c անգամ մուտքագրել x թիվ, և պարզել պատկանում է արդյոք $x \& (\sim x \ll 2)$ գործողության արդյունքը (a, b) միջակայքին: Պատկանելիության դեպքում արտածել արդյունքը:

Խնդիր 13

Մուտքագրել a , b , c բնական թվերը: c անգամ մուտքագրել x թիվը, և պարզել պատկանում է արդյոք $x \& (\sim x \wedge x)$ գործողության արդյունքը (a, b) միջակայքին: Պատկանելիության դեպքում արտածել արդյունքը:

Դաս 7. break, continue օպերատորներ

Ներկայացվում է *while* և *do while* ցիկլում *break*, *continue* հրամանների կիրառությունը համադրած ճյուղավորման օպերատորների հետ: Ներքևում բերվում է *break* հրամանի օգտագործման օրինակ:

```
// Program to calculate the sum of numbers
// If the user enters a negative number, the loop terminates
#include <stdio.h>

int main(void)
{
    double number, sum = 0.0;

    while (1) {
        printf("Enter n ");
        scanf("%lf", &number);

        sum += number; // sum = sum + number;
        // if the user enters a negative number, break the loop
        if (number < 0.0) {
            break;
        }
    }

    printf("Sum = %.2lf", sum);
}
```

Հաջորդ օրինակը ցույց է տալիս *continue* հրամանի կիրառությունը:

```
// Program to calculate the sum of numbers (10 numbers max)
// If the user enters a negative number, it's not added to the result
#include <stdio.h>
int main(void)
{
    int i = 0;
    double number, sum = 0.0;

    while (i < 10) {
```

```

printf("Enter a n%d: ", i);
scanf("%lf", &number);

if (number < 0.0) {
    continue;
}
sum += number; // sum = sum + number;
}

printf("Sum = %.2lf", sum);
}

```

Խնդիր 1

Մուտքագրել թվերի հաջորդականություն և հաշվել դրանցից դրականների քառակուսիների գումարը, քանի դեռ այն մեծ չէ 200-ից:

Խնդիր 2

Մուտքագրել թվերի հաջորդականություն և հաշվել դրանցից բացասականների քառակուսիների արտադրյալը, քանի դեռ այն մեծ չէ 400-ից:

Խնդիր 3

Մուտքագրել ամբողջ թվերի հաջորդականություն՝ մինչև 0-ի հանդիպելը: Արտածել "Yes", եթե մուտքագրված հաջորդականությունը աճող է, հակառակ դեպքում "No":

Խնդիր 4

Մուտքագրել ամբողջ թվերի հաջորդականություն՝ մինչև 0-ի հանդիպելը: Արտածել "Yes", եթե մուտքագրված հաջորդականությունում իրար հավասար երկու հարևան թվերի գոնե մեկ զույգ կա, հակառակ դեպքում "No":

Խնդիր 5

Մուտքագրել ամբողջ թվերի հաջորդականություն, ավարտել ցիկլը, եթե ընթացիկ մուտքագրված թիվը նախորդից փոքր է:

Խնդիր 6

Ստուգել արդյոք մուտքագրված թիվը հանդիսանում է 3-ի աստիճան:

Խնդիր 7

Մուտքագրված $n > 0$ թվի համար հաշվել ամենամեծ երկուսի աստիճան հանդիսացող թիվը, որը չի գերազանցում n -ը:

Խնդիր 8

Մուտքագրված $n > 0$ թվի համար հաշվել ամենափոքր երկուսի աստիճան հանդիսացող թիվը, որը մեծ է n -ից:

Դաս 8. for ցիկլ, համեմատությունը while ցիկլի հետ

for ցիկլի կառուցվածքը

Բերվում է *for* ցիկլի կառուցվածքը և դրանում համեմատման օպերատորների (<, >, ==, !=, >=, <=) օգտագործումը: Ներքևում բերվում է *for* ցիկլի պարզ օրինակ:

```
#include <stdio.h>

int main(void)
{
    for (int i = 0; i < 10; i += 1) {
        printf("%d\n", i);
    }
    printf("%d\n", i);
}
```

Բերվում է *for* ցիկլում *break*, *continue* հրամանների օգտագործման օրինակներ:

for և while ցիկլերի համարժեքություն

Ներկայացվում է *for* և *while* ցիկլերի համարժեքությունը:

<pre>for (statement1; expression; statement2) { statement1; statement1; ... }</pre>	<pre>statement1; while (expression) { statement1; statement1; ... statement2; }</pre>
---	---

Խնդիր 1

Հաշվել տրված $[A, B]$ միջակայքին պատկանող բոլոր ամբողջ թվերի գումարը/արտադրյալը:

Խնդիր 2

Մուտքագրել i բնական թիվը, եթե այն $[10, 20]$ միջակայքից է, ապաարտածել $[i, 20]$ միջակայքի բոլոր ամբողջ թվերը:

Խնդիր 3

Մուտքագրել i բնական թիվը, եթե այն $[0, 100]$ միջակայքից է, ապաարտածել $[0, i]$ միջակայքի բոլոր ամբողջ թվերը անման/նվազման կարգով:

Խնդիր 4

Մուտքագրել N բնական թիվը և օգտագործելով մեկ ցիկլ հաշվել $1! + 2! + 3! + \dots + N!$ Արտահայտության արժեքը:

Խնդիր 5

Մուտքագրել N բնական թիվը և արտածել Ֆիբոնաչիի հաջորդականության առաջին N անդամները:

Խնդիր 6

Ստուգել արդյոք մուտքագրված բնական թիվը պարզ է:

Խնդիր 7

Արտածել 100-ից փոքր բոլոր բնական թվերը, որոնց քառակուսիներըսիմետրիկ են:

Խնդիր 8

Մուտքագրել N բնական թիվը, այնուհետև մուտքագրել N բնականթվեր: Արտածել բոլոր այն թվերը, որոնք պարզ են:

Դաս 9. Փոփոխականների հասանելիության տիրույթ, զանգվածներ

Փոփոխականների հասանելիության տիրույթ

Ներկայացվում են փոփոխականների հասանելիության տիրույթը, գլոբալ և լոկալ փոփոխականները: Ներքևում բերվում է գլոբալ փոփոխականի կիրառման օրինակ:

```
#include <stdio.h>
/* global variable declaration */
int g;
int main(void)
{
    /* local variable declaration */
    int a, b;
    /* actual initialization */
    a = 10;
    b = 20;
    g = a + b;
    printf ("value of a = %d, b = %d and g = %d\n", a, b, g);
}
```

Միաչափ զանգվածներ

Ներկայացվում են զանգվածներ, դրանց կառուցվածքը և ներկայացումը հիշողության մեջ: Ներքևում բերվում են զանգվածի օրինակներ:

```
#include <stdio.h>

int main(void)
{
    int a[10];
```

```

short s[5];
long long_array[51];
float float_array[6];
double double_array[9];
printf("Size of a = %lu\n", sizeof(a));
printf("Size of s = %lu\n", sizeof(s));
printf("Size of double_array = %lu\n", sizeof(double_array));
}

```

Ներքևում բերվում է զանգվածի էլեմենտին դիմելու օրինակ:

```

#include <stdio.h>

int main(void)
{
    int a[5];
    a[0] = 5;
    a[1] = 7;
    a[2] = 89;
    a[3] = 8;
    a[4] = 21;
}

```

Խնդիր 1

Տրված զանգվածում գտնել դրական/բացասական և զույգ/կենտ էլեմենտների քանակը:

Խնդիր 2

Տրված զանգվածում գտնել այն էլեմենտների քանակը, որոնք փոքր/մեծ են իրենց ինդեքսի քառակուսուց:

Խնդիր 3

Արտածել տվյալ զանգվածի բոլոր էլեմենտները, դրանցից հանելով զանգվածի էլեմենտների միջին թվաբանականը:

Խնդիր 4

Տրված զանգվածում գտնել այն էլեմենտների քանակը, որոնք մեծ/փոքր են երկու հարևանների կիսագումարից:

Խնդիր 5

Տրված զանգվածում տեղերով փոխել առաջին մեծագույն և վերջին փոքրագույն էլեմենտները:

Խնդիր 6

Տրված զանգվածում հաշվել բոլոր այն էլեմենտների գումարը, որոնք գտնվում են առաջին մեծագույն և վերջին փոքրագույն էլեմենտների միջև:

Խնդիր 7

Տրված զանգվածում հաշվել կրկնվող տարրերի քանակը:

Խնդիր 8

Հաշվել զանգվածի այն տարրերի քանակը, որոնք հանդիպում են ճիշտ անգամ, որտեղ k-ն տրված բնական թիվ է:

Խնդիր 9

Տրված զանգվածում հաշվել որևէ թվի ֆակտորիալ հանդիսացող էլեմենտների միջին թվաքանականը:

Խնդիր 10

Տրված զանգվածում հաշվել հարևան էլեմենտների ամենամեծ ընդհանուր բաժանարարների արտադրյալը:

Խնդիր 11

Տրված զանգվածում հաշվել հարևան էլեմենտների ամենափոքր ընդհանուր բազմապատիկների գումարը:

Խնդիր 12

Տրված զանգվածում հաշվել այն էլեմենտների քանակը, որոնք բավարարում են $2^k < a^k < k!$ պայմանին, որտեղ k -ն էլեմենտի ինդեքսն է:

Խնդիր 13

Տրված զանգվածից հեռացնել բոլոր 0-ներն ու խտացնել (վերջում0-ներ) առանց օժանդակ զանգված օգտագործելու:

Խնդիր 14

Տրված է a_0, \dots, a_{n-1} հաջորդականությունը: Կառուցել b_0, \dots, b_{n-1} հաջորդականությունը, որտեղ

1. b_i -ն a_i – ի բաժանարարների գումարն է,
2. b_i -ն a_0, \dots, a_i թվերի միջին թվաբանականն է,
3. b_i -ն a_i, \dots, a_0 թվերի ամենափոքր ընդհանուր բազմապատիկն է,
4. b_i -ն a_i թվի պարզ բաժանարարների քանակն է,
5. b_i -ն a_i թվի փոքրագույն թվանշանն է,
6. $b_i=1$, եթե a_0, \dots, a_i -ում կան իրար հավասար էլեմենտներ, $b_i=0$ ՝ հակառակ դեպքում,
7. $b_i = \text{LCF}(a_0, \dots, a_i)$,
8. b_i -ն a_i -ում գրոյական բիթերի քանակն է,
9. b_i -ն a_i թիվը չգերազանցող պարզ թվերի քանակն է,
10. b_i -ն a_i թիվը չգերազանցող բոլոր սիմետրիկ թվերի քանակն է,
11. $b_i = a_i!$:

Խնդիր 15

Տրված է a_0, \dots, a_n հաջորդականությունը: Կառուցել b_0, \dots, b_k ($k \leq n$) ենթահաջորդականությունը a հաջորդականության այն տարրերից, որոնք բավարարում են նշված պայմանին.

1. a_i -ն մեծ է մուտքագրված t թվից,
2. a_i -ն 4-ին պատիկ թիվ է,
3. a_i -ի երկուական ներկայացման մեջ մեկերի քանակը պարզ թիվ է,
4. a_i -ի բաժանարարների քանակը մեծ է կամ հավասար մուտքագրված t թվից,
5. a_i -ի առաջին և վերջին բայթերում մեկերի քանակները հավասար են,
6. a_i -ն սիմետրիկ թիվ է,
7. a_i -ն կատարյալ թիվ է,
8. $a_i = i!$,

Խնդիր 16

Տրված զանգվածում գտնել ինվերսիաների ($i < j \ \&\& \ a[i] > a[j]$) քանակը:

Խնդիր 17

Հայտնի է, որ տրված զանգվածում բոլոր թվերը կրկնվում են ճիշտ երկու անգամ բացի մեկից: Գտնել այդ թիվը:

Խնդիր 18

Տրված են երկու զանգվածներ: Պարզել արդյոք առաջին զանգվածը հանդիսանում է երկրորդի ենթազանգված, թե ոչ:

Խնդիր 19

Տրված են երկու զանգվածներ: Պարզել արդյոք տրված զանգվածիբոլոր էլեմենտները հանդիպում են երկրորդում թե՞ ոչ:

Խնդիր 20

Գտնել տրված զանգվածում ամենաերկար ածող անընդհատ ենթահաջորդականության երկարությունը:

Խնդիր 21

Գտնել տրված զանգվածում ամենաերկար ածող անընդհատ ենթահաջորդականությունը և արտածել:

Դաս 10. Չանգվածների սորտավորում

Ներկայացվում է զանգվածի սորտավորման սկզբունքները, ինչպես նաև պղպջակների և տեղադրմամբ սորտավորման ալգորիթմների նկարագրերը: Իրականացվում է պղպջակների սորտավորման ալգորիթմը, որը բերված է ներքևում:

```
// Implementation of Bubble sort
#include <stdio.h>
int main(void)
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    int i, j, temp;
    for (i = 0; i < n-1; i++) {
        // Last i elements are already in place
        for (j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    printf("Sorted array: \n");
    for (int i=0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}
```

Խնդիր 1

Տրված զանգվածում գտնել կրկնվող էլեմենտների քանակը սորտավորման կիրառմամբ:

Խնդիր 2

Տրված զանգվածում գտնել ամենաշատ կրկնվող էլեմենտը սորտավորման կիրառմամբ:

Խնդիր 3

Տրված են երկու աճման/նվազման կարգով սորտավորված զանգվածներ: Ստեղծել նոր զանգված՝ միաձուլելով երկու տրված զանգվածները այնպես, որ արդյունքը լինի սորտավորված:

Դաս 11. Ցուցիչներ

Ներկայացվում են ցուցիչները և դրանց կիրառության օրինակները, որոնցից մեկը բերված է ներքևում:

```
#include <stdio.h>
int main(void)
{
    int* pc, c;
    c = 22;
    printf("Address of c: %p\n", &c);
    printf("Value of c: %d\n\n", c); // 22
    pc = &c;
    printf("Address of pointer pc: %p\n", pc);
    printf("Content of pointer pc: %d\n\n", *pc); // 22
    c = 11;
    printf("Address of pointer pc: %p\n", pc);
    printf("Content of pointer pc: %d\n\n", *pc); // 11
    *pc = 2;
    printf("Address of c: %p\n", &c);
    printf("Value of c: %d\n\n", c); // 2
}
```

Ներկայացվում է ցուցիչների ապահասցեավորման (dereference) գործողությունը, որոնց օրինակներից մեկը բերվում է հաջորդիվ:

```
int main(void)
{
    /* Dereferencing is used to access or manipulate data contained in
    memory location pointed to by a pointer. */
    int a = 7, b ;
    int *p; // Uninitialized Pointer
    p = &a; // Stores address of a in ptr
    b = *p; // Put value pointed by ptr into b
}
```

Ներկայացվում են հաստատուն ցուցիչները, ցուցիչները հաստատուն արժեքների վրա և հաստատուն ցուցիչները հաստատուն արժեքների վրա: Ներքևում բերվում են համապատասխան օրինակները:

```
#include <stdio.h>

int main(void)
{
    int var1 = 0, var2 = 0; // We declared two variables var1 and
var2
    // A constant pointer 'ptr' is declared and made to point var1
    int *const ptr = &var1;
    ptr = &var2; // wrong! ptr is made to point var2.
    printf("%d\n", *ptr);
}
```

```
#include <stdio.h>

int main(void)
{
    int var1 = 0; // We defined a variable var1 with value 0
    // we defined a pointer to a constant which points to variable
var1
    const int* ptr = &var1;
    *ptr = 1; // wrong! through this pointer we tried to change the
value of var1
    printf("%d\n", *ptr);
}
```

```
#include<stdio.h>

int main(void)
{
    int var1 = 0,var2 = 0; // We declared two variables var1 and
var2.
```

```
// We declared a constant pointer to a constant and made it to
point to var1
const int* const ptr = &var1;
// Now in the next two lines we tried to change the address and
value pointed by the pointer.
*ptr = 1; // wrong
ptr = &var2; // wrong
printf("%d\n", *ptr);
}
```

Խնդիր 1.

Գրել ծրագիր, որը ներմուծված զանգվածի էլեմենտները կշռի՝
օգտագործել ցուցիչներ:

Խնդիր 2.

Գրել ծրագիր, որը ներմուծված երկու տողերը կհամեմատի
այբբենական կարգով:

Դաս 12. Ֆունկցիաներ

Ներկայացվում են ֆունկցիաներ, դրանց կառուցվածքը և կիրառության հիմնական նպատակները: Ներկայացվում են ֆունկցիային արգումենտներ փոխանցելու ձևերը: Ներքևում բերվում է օրինակ:

```
#include <stdio.h>

double circle_area(double radius) {
    return 3.14 * radius * radius;
}

int main(void)
{
    double radius;
    scanf("%lf", &radius);
    double result = circle_area(radius);
    printf("Cycle area is %lf", result);
}
```

Ներկայացվում է զանգվածի փոխանցումը ֆունկցիային, որի օրինակներից մեկը բերված է ներքևում:

```
#include <stdio.h>

float calculateSum(float age[], int size);
int main(void)
{
    float result, age[] = {23.4, 55, 22.6, 3, 40.5, 18};
    // age array is passed to calculateSum()
    result = calculateSum(age, 6);
    printf("Result = %.2f", result);
}

float calculateSum(float age[], int size) {
    float sum = 0.0;
    for (int i = 0; i < size; ++i) {
        sum += age[i];
    }
}
```

```
return sum;
}
```

Խնդիր 1

Գրել ֆունկցիա, որը վերադարձնում է արգումենտում տրված ամբողջթվերի միջին թվաբանականը:

Խնդիր 2

Գրել ֆունկցիա, որն արգումենտում ստանում է իրական թվերի a_0, \dots, a_n հաջորդականություն, k իրական թիվ և վերադարձնում է $P_n(x) = a_0 \cdot x^n + a_1 \cdot x^{n-1} + \dots + a_n$ բազմանդամի արժեքը k կետում:

Խնդիր 3

Գրել ֆունկցիա, որն արգումենտում ստանում է ամբողջ թվերի զանգված և պարզում, զանգվածի մեծագույն տարրը սիմետրիկ է, թե ոչ:

Խնդիր 4

Գրել ֆունկցիա, որը վերադարձնում է արգումենտում ստացված ամբողջ թվերի զանգվածում առաջին բացասական էլեմենտի ինդեքսը և -1 , եթե բացասական էլեմենտ չի պարունակվում:

Խնդիր 5

Գրել ֆունկցիա, որը վերադարձնում է *true*, եթե արգումենտում ստացված բնական թվերի զանգվածում կա գոնե մեկ պարզ թիվ:

Խնդիր 6

Գրել ֆունկցիա, որն արգումենտում ստանում է n բնական թիվը և վերադարձնում է ամենամեծ թվանշանի կարգահամարը, եթե այդ թվանշան թվում հանդիպում է մի քանի անգամ վերադարձնել ամենաձախիկ կարգահամարը:

Խնդիր 7

Գրել ֆունկցիա, որն արգումենտում ստանում է n բնական թիվը և վերադարձնում է այդ թվի բոլոր պարզ բաժանարարների քանակը:

Խնդիր 8

Գրել ֆունկցիա, որն արգումենտում ստանում է n բնական թիվը և հաշվում է, թե քանի անգամ է նրանում հանդիպում առաջին թվանշանը:

Խնդիր 9

Գրել ֆունկցիա, որն արգումենտում ստանում է a և b բնական թվերը և վերադարձնում է նրանց

- 1. ամենամեծ ընդհանուր բաժանարարը,**
- 2. ամենափոքր ընդհանուր բազմապատիկը:**

Խնդիր 10

Գրել ֆունկցիա, որն արգումենտում ստանում է n բնական թիվը և վերադարձնում է *true* արժեք, եթե թվում թվանշանները դասավորված են չնվազման կարգով, *false*՝ հակառակ դեպքում:

Խնդիր 11

Գրել ֆունկցիա, որն արգումենտում ստանում է n բնական թիվը և վերադարձնում է *true* արժեք, եթե թիվը պարզ է, *false*՝ հակառակ դեպքում:

Խնդիր 12

Գրել ֆունկցիա, որը արգումենտում ստանում է երկու թվերի հասցեները թվերի արժեքները միմյանցով փոխարինում (swap):

Խնդիր 13

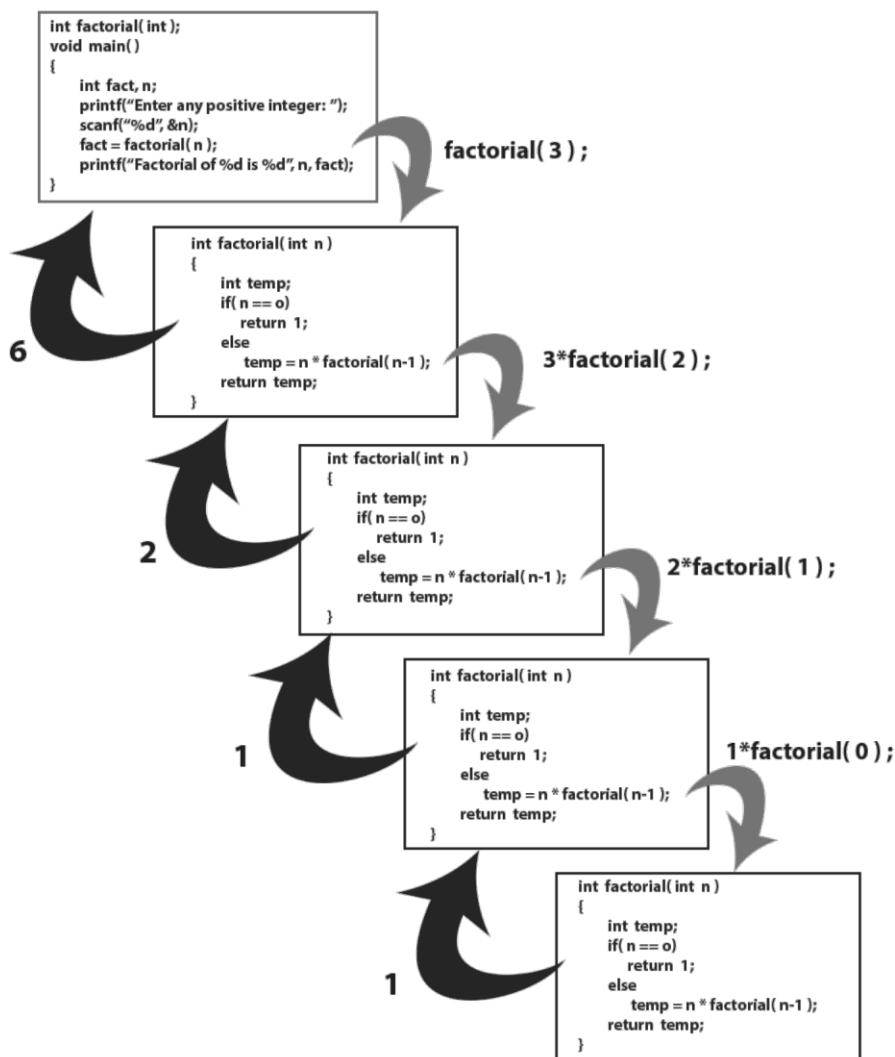
Գրել ֆունկցիա, որը արգումենտում ստանում է զանգված և սիմետրիկ շրջում տարրերը (օգտագործել ցուցիչներ):

Դաս 13. Ռեկուրսիվ ֆունկցիաներ, ձուլմամբ սորտավորում

Ռեկուրսիվ ֆունկցիաներ

Ներկայացվում են ռեկուրսիվ ֆունկցիաները և ռեկուրսիայի հիմնական սկզբունքները: Ներքևում բերվում է ռեկուրսիվ ֆունկցիայի աշխատանքի օրինակ:

Նկարը վերցված է [այստեղից](#) [1]:



Ձուլմամբ սորտավորում

Ներկայացվում է ձուլմամբ սորտավորման (merge sort) ալգորիթմը, որի իրականացումը բերված է ներքևում:

```
#include <stdio.h>
#include <stdlib.h>

// Merges two subarrays of arr[].
// First subarray is arr[l..m]
// Second subarray is arr[m+1..r]
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    // Create temp arrays
    int L[n1], R[n2];

    // Copy data to temp arrays
    // L[] and R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    // Merge the temp arrays back
    // into arr[l..r]
    // Initial index of first subarray
    i = 0;

    // Initial index of second subarray
    j = 0;

    // Initial index of merged subarray
    k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
```

```

        arr[k] =
        L[i];i++;
    }
    else {
        arr[k] = R[j];
        j++;
    }
    k++;
}

// Copy the remaining elements
// of L[], if there are any
while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
}

// Copy the remaining elements of
// R[], if there are any
while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
}
}

// l is for left index and r is
// right index of the sub-array
// of arr to be sorted
void mergeSort(int arr[], int l, int r)
{
    if (l < r) {
        // Same as (l+r)/2, but avoids
        // overflow for large l and r
        int m = l + (r - l) / 2;

        // Sort first and second halves
        mergeSort(arr, l, m);

```

```

        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}

// UTILITY FUNCTIONS
// Function to print an array
void printArray(int A[], int size)
{
    for (int i = 0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

// Driver code
int main(void)
{
    int arr[] = { 12, 11, 13, 5, 6, 7 };
    int arr_size = sizeof(arr) / sizeof(arr[0]);

    printf("Given array \n");
    printArray(arr, arr_size);

    mergeSort(arr, 0, arr_size - 1);

    printf("\nSorted array \n");
    printArray(arr, arr_size);
}

```

Իրականացումը վերցված է [այստեղից](#) [2]:

Խնդիր 1

Գրել ռեկուրսիվ ֆունկցիա, որը հաշվում է արգումենտում տրված $n \geq 0$ ամբողջ թվի ֆակտորիալը:

Խնդիր 2

Գրել ռեկուրսիվ ֆունկցիա, որը վերադարձնում է արգումենտում ստացած բնական թվի թվանշանների գումարը:

Խնդիր 3

Գրել ռեկուրսիվ ֆունկցիա, որը վերադարձնում է արգումենտում ստացած բնական թվի ամենամեծ թվանշանը:

Խնդիր 4

Գրել ռեկուրսիվ ֆունկցիա, որը վերադարձնում է արգումենտում ստացած երկու բնական թվերի ամենամեծ ընդհանուր բաժանարարը:

Խնդիր 5

Գրել ռեկուրսիվ ֆունկցիա, որը կհաշվի $1 + 1/2 + 1/3 + 1/4 \dots + 1/N$ արտահայտության արժեքը, մուտքագրված N բնական թվի համար:

Խնդիր 6

Գրել ռեկուրսիվ ֆունկցիա, որը կհաշվի $1 - 1/2 + 1/3 - 1/4 \dots 1/N$ արտահայտության արժեքը, մուտքագրված N բնական թվի համար:

Դաս 14. Տողեր, բազմաչափ զանգվածներ

Տողեր

Ներկայացվում են տողերը և դրանց հետ հիմնական գործողությունները:
Չաջորդիվ բերվում է տողերի հայտարարման և օգտագործման մի քանի օրինակներ:

```
#include <stdio.h>

int main(void)
{
    // You can initialize strings in a number of ways.
    char c1[] = "abcd";
    char c2[50] = "abcd";
    char c3[] = {'a', 'b', 'c', 'd', '\0'};
    char c4[5] = {'a', 'b', 'c', 'd', '\0'};

    /* Here, we are trying to assign 6 characters (the last character
    is '\0') to a char array having 5 characters. This is bad and you
    should never do this. */

    char c5[5] = "abcde";
    /* Arrays and strings are second-class citizens in C; they do not
    support the assignment operator once it is declared.*/
    char c6[100];
    c6 = "C programming"; // Error! array type is not assignable.
}
```

Ներքևում բերվում է տողի մուտքագրման և արտածման օրինակ:

```
#include <stdio.h>
int main(void)
{
    // scanf() to read a string
    /* The scanf() function reads the sequence of characters until it
    encounters whitespace (space, newline, tab, etc.) */
```

```

char name[20];
printf("Enter name: ");
scanf("%s", name);
printf("Your name is %s.", name);
}

```

Ներքևում բերվում է տողերի մուտքագրման և արտածման այլընտրանքային միջոցի օրինակ:

```

#include <stdio.h>
int main(void)
{
    /* You can use the fgets() function to read a line of string.
    And, you can use puts() to display the string. */
    char name[30];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin); // read string
    printf("Name: ");
    puts(name); // display string
}

```

Դիտարկվում են նաև տողերի հետ աշխատող հետևյալ ֆունկցիաները.

1. *strlen()* - հաշվում է տողի երկարությունը,
2. *strcpy()* - պատճենում է մի տողը մյուսի մեջ,
3. *strcmp()* - համեմատում է երկու տողեր,
4. *strcat()* - միավորում է երկու տողեր:

Բազմաչափ զանգվածներ

Ներկայացվում են բազմաչափ զանգվածները և դրանց հետ հիմնական գործողությունները: Ներքևում բերվում է բազմաչափ զանգվածի օրինակներ:

```

// C program to store temperature of two cities in a week and display it.
#include <stdio.h>

```

```

int main(void)
{
    const int CITY = 2;
    const int WEEK = 7;
    int temperature[CITY][WEEK];
    // Using nested loop to store values in a 2d array
    for (int i = 0; i < CITY; ++i) {
        for (int j = 0; j < WEEK; ++j) {
            printf("City %d, Day %d: ", i + 1, j + 1);
            scanf("%d", &temperature[i][j]);
        }
    }
    printf("\nDisplaying values: \n\n");
    // Using nested loop to display values of a 2d array
    for (int i = 0; i < CITY; ++i) {
        for (int j = 0; j < WEEK; ++j) {
            printf("City %d, Day %d = %d\n", i + 1, j + 1,
temperature[i][j]);
        }
    }
}

```

Հաջորդ օրինակը ցույց է տալիս երկչափ զանգվածի փոխանցումը ֆունկցիային:

```

//Passing two-dimensional arrays
#include <stdio.h>

void displayNumbers(int num[][2], int m);

int main(void)
{
    int num[2][2];
    printf("Enter 4 numbers:\n");
    for (int i = 0; i < 2; ++i)
        for (int j = 0; j < 2; ++j)
            scanf("%d", &num[i][j]);
}

```

```

// passing multi-dimensional array to a function
displayNumbers(num, 2);
}

void displayNumbers(int num[][2], int m)
{
    printf("Displaying:\n");
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < 2; ++j) {
            printf("%d\n", num[i][j]);
        }
    }
}
}

```

Խնդիր 1

Մուտքագրել մինչև 60 սիմվոլներից բաղկացած տեքստ: Արտածել միայն տեքստի մեջ մտնող՝

1. փոքրատառերը,
2. մեծատառերը,
3. կետադրական նշանները:

Խնդիր 2

Մուտքագրել մինչև 40 սիմվոլ պարունակող տեքստ, որն ավարտվում է բացատով: Հեռացնել նրանից բոլոր ստորակետերը և արտածել ստացված տեքստը:

Խնդիր 3

Մուտքագրել մինչև 40 սիմվոլ պարունակող տեքստ, որն ավարտվում է բացատով: Տողում յուրաքանչյուր "abc" ենթատող փոխարինել "def"-ով և արտածել ստացված տեքստը:

Խնդիր 4

Մուտքագրել մինչև 40 սիմվոլ պարունակող տեքստ, որն ավարտվում է բացատով: Չեռացնել նրանից առաջին հանդիպող 'w' սիմվոլը, եթե այդպիսինկա, և արտածել ստացված տեքստը:

Խնդիր 5

Մուտքագրել մինչև 40 սիմվոլ պարունակող տեքստ, որն ավարտվում է բացատով: Արտածել տեքստում յուրաքանչյուր սիմվոլ հայտնվելու քանակը:

Խնդիր 6

Տրված մատրիցայի համար արտածել դրա գլխավոր և երկրորդական անկյունագծերի էլեմենտները:

Խնդիր 7

Տրված մատրիցայի համար արտածել դրա գլխավոր և երկրորդական անկյունագծերի էլեմենտների գումարը/արտադրյալը:

Խնդիր 8

Տրված մատրիցայի համար հաշվել դրա դետերմինանտը:

Խնդիր 9

Տրված երկու համապատասխան չափի մատրիցաների համար հաշվել դրանց`

- 1. գումար,**
- 2. տարբերությունը,**
- 3. արտադրյալը:**

Դաս 15. Դինամիկ զանգվածներ

Ներկայացվում են դինամիկ զանգվածները և դրանց հետ աշխատանքի հիմնական սկզբունքները: Ներկայացվում են *malloc()* և *free()* ֆունկցիաները: Ներքևում բերվում է կիրառության օրինակ:

```
// Program to calculate the sum of n numbers entered by the user
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) malloc(n * sizeof(int));
    /* The above statement allocates n*4 bytes of memory. It's because
    the size of int is 4 bytes. And, the pointer ptr holds the address of
    the first byte in the allocated memory. The expression results in a
    NULL pointer if the memory cannot be allocated. */
    // if memory cannot be allocated
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(1);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);
    // deallocating the memory
    free(ptr);
}
```

Դինամիկ բազմաչափ զանգվածներ

Ներկայացվում են դինամիկ բազմաչափ զանգվածները և դրանց հետ աշխատանքի հիմնական սկզբունքները:

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int N, M;
    scanf("%d%d", &N, &M);
    int **A = (int **)malloc(N*sizeof(int *));
    for(int i = 0; i < N; i++) {
        A[i] = (int *)malloc(M*sizeof(int));
    }

    for(int i = 0; i < N; i++)
        for(int j = 0; j < M; j++)
            A[i][j] = i;
    for(int i = 0; i < N; i++) {
        for(int j = 0; j < M; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }
    for(int i = 0; i < N; i++) {
        free(A[i]);
    }
    free(A);
}
```

Խնդիր 1

Ներմուծել N, K, M բնական թվերը, ապա N x K և K x M չափերով մատրիցները:
Հաշվել դրանց արտադրյալը:

Խնդիր 2

Ներմուծել N , M բնական թվերը, ապա $N \times M$ չափի մատրիցա և գրոյով փոխարինել մատրիցի այն տողերը, որոնց տարրերը համընկնում են առաջին տողի տարրերի հետ:

Խնդիր 3

Ներմուծել N , M բնական թվերը, ապա $N \times M$ չափի մատրիցա և տողերը կարգավորել նրանցում դրական թվերի գումարների աճման կարգով:

Խնդիր 4

Ներմուծել N բնական թվերը, ապա $N \times N$ չափի մատրիցա և ստեղծել նոր մատրիցա, որը ներմուծվածի տրանսպոնացվածն է:

Խնդիր 5

Ստեղծել ստեկ (*stack*) տվյալների կառուցվածքը:

Խնդիր 6

Ստեղծել հերթ (*queue*) տվյալների կառուցվածքը:

Դաս 16. Կառուցվածքներ (structures)

Ներկայացվում են
դրանց
կառուցվածքները,
կիրառությունները
և
հատկությունները: Ստորև բերվում է օրինակ:

```
#include <stdio.h>

struct Point {
    int x;
    int y;
};

int main() {
    struct Point p1 = {1, 2};
    struct Point p2 = {2, 3};
    printf("(%d,%d)\n", p1.x, p1.y);
    printf("(%d,%d)\n", p2.x, p2.y);
}
```

Նաև ներկայացվում են
կառուցվածքը
հնչպես
կարելի է
փոխանցել
ֆունկցիային:

```
#include <stdio.h>

struct Point {
    int x;
    int y;
};

void printQuadrant(struct Point p) {
    if (p.x > 0 && p.y > 0) {
        printf("First quadrant\n");
        return;
    }
    if (p.x < 0 && p.y > 0) {
        printf("Second quadrant\n");
        return;
    }
    if (p.x < 0 && p.y < 0) {
```

```

    printf("Third quadrant\n");
    return;
}
if (p.x > 0 && p.y < 0) {
    printf("Forth quadrant\n");
    return;
}
if (p.x == 0) {
    printf("point is in y\n");
}

if (p.y == 0) {
    printf("point is in x\n");
}
}

int main() {
    struct Point p1 = {1, 2};
    printQuadrant(p1);
}

```

Խնդիր 1

Ստեղծել միակապ ցուցակ տվյալների կառուցվածքը:

Խնդիր 2

Ստեղծել երկկապ ցուցակ տվյալների կառուցվածքը:

Խնդիր 3

Ստեղծել կառուցվածք՝ ժամանակը ներկայացնելու համար՝ ժամերի, րոպեների և վայրկյանների համար նախատեսված դաշտերով: Գրել ֆունկցիաներ՝ ժամանակի արժեքները ավելացնելու կամ հանելու, 12-ժամյա և 24-ժամյա ձևաչափերի փոխակերպման և երկու ժամանակալին արժեքների միջև ժամանակի տարբերությունը հաշվարկելու համար:

Խնդիր 4

Ստեղծել կառուցվածք ուսանողին ներկայացնելու համար՝ անվան, ID-ի դաշտերով և թեստերի միավորների գանգվածով: Իրականացնել ֆունկցիաներ թեստերի միջին միավորը հաշվարկելու, ամենաբարձր և ամենացածր միավորները գտնելու և ուսանողի թեստի միավորները թարմացնելու համար:

Խնդիր 5

Ստեղծել կառուցվածք շրջանը ներկայացնելու համար՝ կենտրոնի x , y կոորդինատների դաշտերով և շառավղով: Իրականացնել ֆունկցիաներ՝ շրջանի շրջագծի երկարությունը, մակերեսը և տրամագիծը հաշվարկելու համար:

Երախտագիտություն

Շնորհակալություն ենք հայտնում Հայ-Ռուսական համալսարանի, մաթեմատիկայի և ինֆորմատիկայի ինստիտուտի համակարգային ծրագրավորման ամբիոնի այն բոլոր դասախոսներին, ովքեր օգնել են խնդիրների ստեղծման, հավաքագրման, ինչպես նաև ճշգրտման հարցերում:

Գրականության ցանկ

1. Brian W.Kernighan, Dennis M.Ritchie, The C Programming Language, Second Edition, 1988.
2. Ռեկուրսիվ ֆունկցիայի աշխատանքի սխեմատիկ նկարագրություն՝ http://www.btechsmartclass.com/c_programming/C-Recursive-Functions.html
3. Չուլմամբ սորտավորման իրականացում C լեզվով՝ <https://www.geeksforgeeks.org/c-program-for-merge-sort/>

Նախաբան.....	6
Դաս 1. Ծանոթություն C լեզվի հետ	7
Ինդիր 1	7
Ինդիր 2	8
Ինդիր 3	8
Ինդիր 4	8
Ինդիր 5	8
Դաս 2. Տիպեր, մոդիֆիկատորներ, գործողություններ, սահմաններ, տվյալների մուտքագրում և ելքագրում	9
Ամբողջ տիպերի մոդիֆիկատորներ, հիմնական գործողություններ.....	9
Ամբողջ տիպի փոփոխականների սահմաններ.....	10
Առանց նշանի թվեր	11
Տվյալների մուտքագրում.....	11
Սահող կետով թվեր	11
Ինդիր 1	12
Ինդիր 2	12
Ինդիր 3	13
Ինդիր 4	13
Ինդիր 5	13
Դաս 3. Միմվոլային տիպ, տվյալների կոդավորում, sizeof օպերատոր	15
Միմվոլների կոդավորում	15
sizeof օպերատոր.....	16
Տվյալների կոդավորման սկզբունքներ.....	16
Ինդիր 1	18
Ինդիր 2	18
Ինդիր 3	18
Ինդիր 4	18
Ինդիր 6	19
Դաս 4. Բիթային, համեմատման և ճյուղավորման օպերատորներ	20
Բիթային օպերատորներ.....	20
Ճյուղավորման պարզագույն օպերատոր	20
Համեմատման օպերատորներ	21
Ինդիր 1	23
Ինդիր 2	23
Ինդիր 3	23
Ինդիր 4	23

Իսնդիր 5	23
Իսնդիր 6	24
Իսնդիր 7	24
Իսնդիր 8	24
Իսնդիր 9	24
Իսնդիր 10	24
Գլուխ 5. Ներդրված ճյուղավորման օպերատորներ,switch/case օպերատոր, enum տիպ	25
Ներդրված ճյուղավորման օպերատորներ	25
switch/case օպերատոր.....	26
enum տիպ.....	27
Իսնդիր 1	27
Իսնդիր 2	27
Իսնդիր 3	28
Իսնդիր 4	28
Իսնդիր 5	28
Իսնդիր 6	28
Գլուխ 6. while և do/while ցիկլեր	29
Իսնդիր 1	30
Իսնդիր 2	31
Իսնդիր 3	31
Իսնդիր 4	31
Իսնդիր 5	31
Իսնդիր 6	31
Իսնդիր 7	31
Իսնդիր 8	31
Իսնդիր 9	32
Իսնդիր 10	32
Իսնդիր 11	32
Իսնդիր 12	32
Իսնդիր 13	32
Գլուխ 7. break, continue օպերատորներ	33
Իսնդիր 1	34
Իսնդիր 2	34
Իսնդիր 3	34
Իսնդիր 4	34
Իսնդիր 5	34
Իսնդիր 6	35
Իսնդիր 7	35
Իսնդիր 8	35
Գլուխ 8. for ցիկլ, համեմատությունը while ցիկլի հետ	36
for ցիկլի կառուցվածքը	36

for և while ցիկլերի համարժեքությունը	36
Ինդիր 1	37
Ինդիր 2	37
Ինդիր 3	37
Ինդիր 4	37
Ինդիր 5	37
Ինդիր 6	37
Ինդիր 7	37
Ինդիր 8	37
Դաս 9. Փոփոխականների հասանելիության տիրույթ, զանգվածներ.....	38
Փոփոխականների հասանելիության տիրույթ	38
Միաչափ զանգվածներ	38
Ինդիր 1	39
Ինդիր 2	39
Ինդիր 3	39
Ինդիր 4	40
Ինդիր 5	40
Ինդիր 6	40
Ինդիր 7	40
Ինդիր 8	40
Ինդիր 9	40
Ինդիր 10	40
Ինդիր 11	41
Ինդիր 12.....	41
Ինդիր 13	41
Ինդիր 14	41
Ինդիր 15	42
Ինդիր 16	42
Ինդիր 17	42
Ինդիր 18	42
Ինդիր 19	43
Ինդիր 20	43
Ինդիր 21	43
Դաս 10. Չանգվածների սորտավորում	44
Ինդիր 1	44
Ինդիր 2	45
Ինդիր 3	45
Դաս 11. Ցուցիչներ	46
Ինդիր 1.	48
Ինդիր 2.	48
Դաս 12. Ֆունկցիաներ	49

Իսնդիր 1	50
Իսնդիր 2	50
Իսնդիր 3	50
Իսնդիր 4	50
Իսնդիր 5	50
Իսնդիր 6	50
Իսնդիր 7	51
Իսնդիր 8	51
Իսնդիր 9	51
Իսնդիր 10	51
Իսնդիր 11	51
Իսնդիր 12	51
Իսնդիր 13	51
Դաս 13. Ռեկուրսիվ ֆունկցիաներ, ձուլմամբ սորտավորում	52
Ռեկուրսիվ ֆունկցիաներ	52
Ձուլմամբ սորտավորում	53
Իսնդիր 1	55
Իսնդիր 2	56
Իսնդիր 3	56
Իսնդիր 4	56
Իսնդիր 5	56
Իսնդիր 6	56
Դաս 14. Տողեր, բազմաչափ զանգվածներ	57
Տողեր	57
Բազմաչափ զանգվածներ	58
Իսնդիր 1	60
Իսնդիր 2	60
Իսնդիր 3	60
Իսնդիր 4	61
Իսնդիր 5	61
Իսնդիր 6	61
Իսնդիր 7	61
Իսնդիր 8	61
Իսնդիր 9	61
Դաս 15. Դինամիկ զանգվածներ	62
Դինամիկ բազմաչափ զանգվածներ	63
Իսնդիր 1	63
Իսնդիր 2	64
Իսնդիր 3	64
Իսնդիր 4	64
Իսնդիր 5	64

Ինդիր 6	64
Դաս 16. Կառուցվածքներ (structures)	65
Ինդիր 1	66
Ինդիր 2	66
Ինդիր 3	66
Ինդիր 4	67
Ինդիր 5	67
Երախտագիտություն	68
Գրականության ցանկ	69

Սևակ Սենիկի Սարգսյան
Հայկ Կարենի Ասլանյան

«Ալգորիթմներ և ալգորիթմական լեզուներ»
Գործնական պարապմունքների ուղեցույց

ԳՀԽ գլխավոր խմբագիր – Մ.Է. Ավագյան
Սրբագրիչ – Ա.Ս. Եսայան
Համակարգչային էջադրում – Ա.Ս. Բժիկյան

Հայ-ռուսական (Սլավոնական)
համալսարանի Գիտական
հրատարակումների խմբագրության հասցե՝
0051, ք. Երևան, փող. Հովսեփ Էմին, 123
հեռ./ֆաքս՝ (+374 12) 77-57-75 (ներքին՝ 392)
Էլ. փոստ՝ maria.avakian@rau.am

Ստորագրված է էլեկտրոնային հրատարակման՝ 03.05.2024թ.