MINISTERUL EDUCAȚIEI
AL REPUBLICII MOLDOVA
UNIVERSITATEA TEHNICĂ
A MOLDOVEI

DEPARTAMENTUL INGINERIA
SOFTWARE ŞI AUTOMATICĂ

MINISTRY OF EDUCATION
OF THE REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY
OF MOLDOVA

Disciplina: Programarea Calculatoarelor

Lucrarea de laborator №5

Tema: Lucrul cu funcțiile în C. Definirea unei funcții în C. Argumente și variabile locale. Returnarea rezultatelor funcției. Declararea şi apelarea funcţiilor. Rezolvarea problemelor.

A efectuat student: _____(Țâțu Cristian, IA-203)

A controlat:_____(lec.unv. Guțu Maria)

Chişinău, 2020

Scopul lucrări: Familarizarea cu lucrul cu funcțiile în limbajul C.

Realizarea sarcinilor:

1. De la tastatură se introduc patru numere a, b, c, d. Scrieți un program C ce va determina:
   a. Numărul mai mare dintre ele;
   b. Numărul mai mic dintre ele;
   c. Cel mai mare divizor comun al lor;
   d. Cel mai mic multiplu comun al lor;
   e. Dacă unul dintre numere este divizor pentru toate celelalte;
   f. Dacă unul dintre ele este multiplu al celorlalte numere.

Utilizați funcții pentru fiecare dintre cazuri.

Ex.1

```
#include <stdlib.h>
#include <stdio.h>
int main() {
  int a, b, c, d, cmmdc_global, cmmmc_global;
  printf("Introduceti a,b,c,d:\n");
  scanf("%d %d %d %d", &a, &b, &c, &d);
  printf("\nMinimul din nr. introduse: %d",min(a,b,c,d));
  printf("\nMaximul din nr. introduse: %d",max(a,b,c,d));
  cmmdc_global = cmmdc(a,b,c,d);
  printf("\nCel mai mare divizor comun: %d",cmmdc(a,b,c,d));
  printf("\nCel mai mic multiplu comun: %d",cmmmc(a,b,c,d,cmmdc_global));
  cmmmc_global = cmmmc(a,b,c,d,cmmdc_global);
  if ( divizor(a,b,c,d,cmmdc_global) == cmmdc_global) printf("\nNumarul %d este divizor
pentru toate celelalte numere", divizor(a,b,c,d,cmmdc_global) );
  else printf("\nNu este nici un divizor din input pentru toate celelalte numere");
  if ( multiplu(a,b,c,d,cmmmc_global) == cmmmc_global) printf("\nNumarul %d este
multiplu pentru toate celelalte numere", (a,b,c,d,cmmmc_global) );
  else printf("\nNu este nici un multiplu din input pentru toate celelalte numere");
  return 0;}

int min ( int a, int b, int c, int d){
    int min_local = INT_MAX;
```

```
    if (min_local > a ) min_local = a;
    if (min_local > b ) min_local = b;
    if (min_local > c ) min_local = c;
    if (min_local > d )min_local = d;
    return min_local;
}

int max ( int a, int b, int c, int d){
    int max_local = INT_MIN;
    if (max_local < a ) max_local = a;
    if (max_local < b ) max_local = b;
    if (max_local < c ) max_local = c;
    if (max_local < d ) max_local = d;
    return max_local;
}

int cmmdc ( int a, int b, int c, int d){
while(a!=b)
    {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
while(c!=d)
    {
        if(c>d)
            c=c-d;
        else
            d=d-c;
    }
while(a!=c)
    {
        if(c>d)
```

```
            a=a-c;
        else
            c=c-a;
    }
    return a;
}


int cmmmc ( int a, int b, int c, int d, int cmmdc_global){
    int cmmmc_local;
    cmmmc_local = (a*b*c*d)/cmmdc_global;
    return cmmmc_local;
}

int divizor ( int a, int b, int c, int d, int cmmdc_global) {
    if ( a == cmmdc_global ) cmmdc_global = a;
    else if ( b == cmmdc_global ) cmmdc_global = b;
    else if ( c == cmmdc_global ) cmmdc_global = c;
    else if ( d == cmmdc_global ) cmmdc_global = d;
    return cmmdc_global;
}

int multiplu ( int a, int b, int c, int d, int cmmmc_global) {
    if ( a == cmmmc_global ) cmmmc_global = a;
    else if ( b == cmmmc_global ) cmmmc_global = b;
    else if ( c == cmmmc_global ) cmmmc_global = c;
    else if ( d == cmmmc_global ) cmmmc_global = d;
    else return 0;
    return cmmmc_global;
}
```

2. De la tastatură se introduc patru numere naturale a, b, c, d. Scrieți un program C ce va determina:
    a. Suma cifrelor fiecărui număr;
    b. Cifre cea mai mare din componența fiecărui număr;
    c. Numărul divizorilor fiecărui număr dat;
    d. Suma divizorilor fiecărui număr cu valoarea mai mică decât a numărului dat;
    e. Dacă printre numerele date sunt numere perfecte;
    f. Dacă numărul respectiv este prim sau nu.

Utilizați funcții pentru fiecare dintre cazuri.

Ex.2

```
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
int main() {
  unsigned int a, b, c, d;
  printf("Introduceti a,b,c,d:\n");
  scanf("%u %u %u %u", &a, &b, &c, &d);
  printf("\nSuma cifrelor pentru a = %u este %u", a, suma_cifrelor(a));
  printf("\nSuma cifrelor pentru b = %u este %u", b, suma_cifrelor(b));
  printf("\nSuma cifrelor pentru c = %u este %u", c, suma_cifrelor(c));
  printf("\nSuma cifrelor pentru d = %u este %u", d, suma_cifrelor(d));
  printf("\n\nCea mai mare cifra din numarul a = %u este %u", a, cmm_cifra(a));
  printf("\nCea mai mare cifra din numarul b = %u este %u", b, cmm_cifra(b));
  printf("\nCea mai mare cifra din numarul c = %u este %u", c, cmm_cifra(c));
  printf("\nCea mai mare cifra din numarul d = %u este %u", d, cmm_cifra(d));
  printf("\n\nNumarul divizorilor numarului a = %u este %u", a, count_divizori(a));
  printf("\nNumarul divizorilor numarului b = %u este %u", b, count_divizori(b));
  printf("\nNumarul divizorilor numarului c = %u este %u", c, count_divizori(c));
  printf("\nNumarul divizorilor numarului d = %u este %u\n", d, count_divizori(d));
  if (sum_divizori(a) < a ) printf("\nSuma divizorilor %u este mai mica decat a = %u", a,
sum_divizori(a) );
  else if (sum_divizori(b) < b ) printf("\nSuma divizorilor %u este mai mica decat a = %u", b,
sum_divizori(b) );
  else if (sum_divizori(c) < c ) printf("\nSuma divizorilor %u este mai mica decat a = %u", c,
sum_divizori(c) );
```

```
  else if (sum_divizori(d) < d ) printf("\nSuma divizorilor %u este mai mica decat a = %u", d,
sum_divizori(d) );
  else printf("\n\nNici o suma a divizorilor numerelor nu este mai mica decat numerele
introduse");
  printf("\n\nNumarul a = %u ", a);nr_perfect(a);
  printf("Numarul b = %u ", b);nr_perfect(b);
  printf("Numarul c = %u ", c);nr_perfect(c);
  printf("Numarul d = %u ", d);nr_perfect(d);
  printf("\n\nNumarul a = %u ", a);nr_prim(a);
  printf("\nNumarul b = %u ", b);nr_prim(b);
  printf("\nNumarul c = %u ", c);nr_prim(c);
  printf("\nNumarul d = %u ", d);nr_prim(d);
  printf("\n");
  return 0;
}

int suma_cifrelor (unsigned int n) {
  int sum = 0, m;
  while (n > 0)
  {
    m = n % 10;
    sum = sum + m;
    n = n / 10;
  }
  return sum;
}

int cmm_cifra (unsigned int n) {
  // unsigned int max - 65535 => 5 elemente max
  unsigned int array[5], sum = 0, m, count = 0;
  for ( int i = 0; i < 5; ++i) {
    array[i] = 0;
  }
  while (n > 0)
```

```
  {
    m = n % 10;
    array[count] = m;
    sum = sum + m;
    n = n / 10;
    ++count;
  }
  int max = array[0];
  for ( int i = 0; i < 5; ++i) {
    if ( array[i] > max ) max = array[i];
  }
  return max;
}

int count_divizori (unsigned int x) {
  unsigned int count = 0;
  for (int i = 1; i <= x; ++i) {
    if ((x % i) == 0) {
      //printf("\t%d", i);
      ++count;
    }
  }
  return count;
}

int sum_divizori (unsigned int x) {
  unsigned int sum = 0;
  for (int i = 1; i <= x; ++i) {
    if ((x % i) == 0) {
      //printf("\t%d", i);
      sum += i;
    }
  }
  return sum;
```

```
}

void nr_perfect (unsigned int x) {
  unsigned int rest, sum = 0;
  for (int i = 1; i <= (x - 1); ++i) {
    rest = x % i;
    if (rest == 0)
    {
      sum += i;
    }
  }
  if (sum == x)
    printf("este perfect\n");
  else
    printf("nu este perfect\n");
}

void nr_prim (unsigned int x) {
  bool flag = 0;
  for (int i = 2; i <= (x / 2); ++i)
    if (x % i == 0) {
      flag = 1;
      break;
    }
  if (x == 1) printf("1");
  else {
    if (flag == 0)
      printf("este un numar prim");
    else
      printf("nu este un numar prim");
  }
}
```

Ex.3

3. De la tastatură se introduc patru numere naturale a, b, c, d. Scrieți un program C ce va determina:
   a. Numărul cel mai mare dintre oglinditul numerelor date;
   b. Numărul cel mai mic dintre oglinditul numerelor date;
   c. Numărul cel mai mare format dintre cifrele fiecărui număr respectiv;
   d. Numărul cel mai mic format dintre cifrele fiecărui număr respectiv (în componența numerelor poate fi și cifra zero).

Utilizați funcții pentru fiecare dintre cazuri.

```
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
int main() {
  unsigned int a, b, c, d;
  printf("Introduceti a,b,c,d:\n");
  scanf("%u %u %u %u", &a, &b, &c, &d);
  printf("\nNumarul oglindit cel mai mare este %u ", max(a,b,c,d) );
  printf("\nNumarul oglindit cel mai mic este %u ", min(a,b,c,d) );
  printf("\n\nNumarul cel mai mare format din cifrele numarului a = %u este -
",a);aranjeaza_max(a);
  printf("\nNumarul cel mai mic format din cifrele numarului a = %u este -
",a);aranjeaza_min(a);
  printf("\n\nNumarul cel mai mare format din cifrele numarului b = %u este -
",b);aranjeaza_max(b);
  printf("\nNumarul cel mai mic format din cifrele numarului b = %u este -
",b);aranjeaza_min(b);
  printf("\n\nNumarul cel mai mare format din cifrele numarului c = %u este -
",c);aranjeaza_max(c);
  printf("\nNumarul cel mai mic format din cifrele numarului c = %u este -
",c);aranjeaza_min(c);
  printf("\n\nNumarul cel mai mare format din cifrele numarului d = %u este -
",d);aranjeaza_max(d);
  printf("\nNumarul cel mai mic format din cifrele numarului d = %u este -
",d);aranjeaza_min(d);
  return 0;
```

```
}

unsigned int oglindit (unsigned int n) {
  unsigned int nr_oglindat = 0, rest;
  while (n != 0)
  {
    rest = n % 10;
    nr_oglindat = nr_oglindat * 10 + rest;
    n = n / 10;
  }
  return nr_oglindat;
}

int max (unsigned int a, unsigned int b, unsigned int c, unsigned int d){
    a = oglindit(a);
    b = oglindit(b);
    c = oglindit(c);
    d = oglindit(d);
    unsigned int max_local = 0;
    if (max_local < a ) max_local = a;
    if (max_local < b ) max_local = b;
    if (max_local < c ) max_local = c;
    if (max_local < d ) max_local = d;
    return max_local;
}

int min (unsigned int a, unsigned int b, unsigned int c, unsigned int d){
    a = oglindit(a);
    b = oglindit(b);
    c = oglindit(c);
    d = oglindit(d);
    unsigned int min_local = INT_MAX;
    if (min_local > a ) min_local = a;
    if (min_local > b ) min_local = b;
```

```
    if (min_local > c ) min_local = c;
    if (min_local > d ) min_local = d;
    return min_local;
}

void aranjeaza_max (unsigned int n) {
  // unsigned int max - 65535 => 5 elemente max
  unsigned int array[5], sum = 0, m, count = 0;
  for ( int i = 0; i < 5; ++i) {
    array[i] = 0;
  }
  while (n > 0)
  {
    m = n % 10;
    array[count] = m;
    sum = sum + m;
    n = n / 10;
    ++count;
  }
  for ( int i = 0; i < count; ++i) {
    for ( int j = 0; j < count - i - 1; ++j) {
      if (array[j] < array[j + 1]) {
        unsigned int temp = array[j];
        array[j] = array[j + 1];
        array[j + 1] = temp;
      }
    }
  }
  for (int i = 0; i < count; ++i) printf("%u", array[i]);
}

void aranjeaza_min (unsigned int n) {
  unsigned int array[5], sum = 0, m, count = 0;
  for ( int i = 0; i < 5; ++i) {
```

```
    array[i] = 0;
  }
  while (n > 0)
  {
    m = n % 10;
    array[count] = m;
    sum = sum + m;
    n = n / 10;
    ++count;
  }
  for ( int i = 0; i < count; ++i) {
    for ( int j = 0; j < count - i - 1; ++j) {
      if (array[j] > array[j + 1]) {
        unsigned int temp = array[j];
        array[j] = array[j + 1];
        array[j + 1] = temp;
      }
    }
  }
  for (int i = 0; i < count; ++i) printf("%u", array[i]);
}
```

Ex.4

Problema Săptămânii:
21.11.2020 - 28.11.2020
Se dă matricea A(m,n) ale cărei componente sunt numere întregi. Să se determine toate punctele „ȘA" și poziția lor. Componenta a[i][j] se numește „ȘA"dacă ea este componentă minimală în linia i și componentă maximală în coloana j.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
  unsigned int n, m;
  printf("Introduceti cate linii are tabelul: ");
```

```
scanf("%u", &n);
printf("Introduceti cate coloane are tabelul:");
scanf("%u", &m);
int A[n][m];
printf("Introduceti %d variabile:\n", n * m);
int min_row, min_position_i_row[n], min_position_j_row[n], R[n];
int max_column, max_position_i_column[m], max_position_j_column[m], C[m];
for ( int i = 0; i < n; ++i) {
  min_position_i_row[i] = 0;
  min_position_j_row[i] = 0;
  R[i] = 0;
  for ( int j = 0; j < m; ++j) {
    scanf("%d", &A[i][j]);
    max_position_i_column[j] = 0;
    max_position_j_column[j] = 0;
    C[j] = 0;
  }
}
printf("Valorile tabloului A:\n");
for ( int i = 0; i < n; ++i) {
  for ( int j = 0; j < n; ++j) {
    printf("%4d", A[i][j]);
  }
  printf("\n");
}
// Valorile minimale pe linii:
int min_position_i_row_var = 0,
    min_position_j_row_var = 0;
for ( int i = 0; i < n; ++i) {
  min_row = A[i][0];
  for ( int j = 0; j < m; ++j) {
    if ( A[i][j] < min_row ) {
      min_row = A[i][j];
      min_position_i_row_var = i;
```

```
        min_position_j_row_var = j;
      }
    }
    R[i] = min_row;
    min_position_i_row[i] = min_position_i_row_var;
    min_position_j_row[i] = min_position_j_row_var;
    printf("Elementul cu pozitia A[%d][%d] e minimul liniei %d si are valoarea = %d\n", i,
min_position_j_row[i], i, R[i]);
  }
  int max_position_i_column_var = 0,
      max_position_j_column_var = 0;
  // Valorile maximale pe coloane
  for ( int k = 0; k < m; ++k) {
    max_column = A[0][k];
    for ( int l = 0; l < n; ++l) {
      if ( A[l][k] > max_column ) {
        max_column = A[l][k];
        max_position_i_column_var = l;
        max_position_j_column_var = k;
      }
    }
    C[k] = max_column;
    max_position_i_column[k] = max_position_i_column_var;
    max_position_j_column[k] = max_position_j_column_var;

    printf("Elementul cu pozitia A[%d][%d] e maximul coloanei %d si are valoarea = %d\n",
max_position_i_column[k], k, k, C[k]);
  }


  // Verificarea daca punctul este "Sa"
  unsigned int count = 0;
  if ( n > m ) count = n;
  else count = m;
```

```
  for ( int i = 0; i < count; ++i) {
    if ( (max_position_i_column[i] == min_position_i_row[i]) && (C[i] == R[i])
        && (max_position_j_column[i] == min_position_j_row[i]) )
      printf("Elementul cu pozitia A[%d][%d] = %d este punct 'sa'\n", min_position_i_row[i],
min_position_j_row[i], R[i]);
  }
  return 0;}
```

Concluzie:

La lucrarea de laborator cu nr.5, am efectuat 5 probleme in limbajul C după sarcinile oferite de profesor. Am utilizat diferite biblioteci, instrucțiuni și algoritmi.