

**Laborator #3**

*La Programarea Calculatorului*

**A efectuat: st.gr.IA-203 \_\_\_\_\_\_\_\_Dascal Vasile**

**A verificat : lector.univ\_\_\_\_\_\_\_\_\_\_ Guțu Maria**

**Chișinăn 2020**

**Probleme cu Funcții**

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.

**Program**

#include <stdio.h>

#include <stdlib.h>

int max\_elm(int a, int b, int c, int d){

if(a>c && a>b && a>d){

return a;

}else if(b>a && b>c && b>d){

return b;

}else if(c>a && c>b && c>d){

return c;

}else if(d>a && d>b && d>c){

return d;

}

}

int min\_elm(int a, int b, int c, int d){

if(a<c && a<b && a<d){

return a;

}else if(b<a && b<c && b>d){

return b;

}else if(c<a && c<b && c<d){

return c;

}else if(d<a && d<b && d<c){

return d;

}

}

int divivor(int a, int b, int c, int d){

while(a!=b){

if(a>b) a=a-b;

else b=b-a;

}

while(b!=c){

if(b>c) b=b-c;

else c=c-b;

}

while(c!=d){

if(c>d) c=c-d;

else d=d-c;

}

}

int multiplu\_min(int a, int b, int c, int d){

int v[3],i;

v[0]=a;

v[1]=b;

v[2]=c;

v[3]=d;

int c\_mai\_mic\_mult\_com = v[1];

for ( i = 1; i <3; i++){

int c\_vec1 = c\_mai\_mic\_mult\_com; int copie\_EL = v[i];

while(c\_vec1 != copie\_EL){

if (c\_vec1 > copie\_EL)

c\_vec1 = c\_vec1 - copie\_EL;

else if (copie\_EL > c\_vec1)

copie\_EL = copie\_EL - c\_vec1;

}

c\_mai\_mic\_mult\_com = c\_mai\_mic\_mult\_com\*(v[i]/copie\_EL);

}

return c\_mai\_mic\_mult\_com;

}

int e\_divizor(int a, int b, int c, int d){

if(b%a==0 && c%a==0 && d%a==0 ){

return a;

}else if(a%b==0 && c%b==0 && d%b==0 ){

return b;

}else if(a%c==0 && b%c==0 && d%c==0 ){

return c;

}else if(a%d==0 && b%d==0 && c%d==0 ){

return c;

}

}

int f\_multiplu(inta, int b, int c, int d){

if(b-a==0 && c-a==0 && d-a==0 ){

return a;

}else if(a-b==0 && c-b==0 && d-b==0 ){

return b;

}else if(a-c==0 && b-c==0 && d-c==0 ){

return c;

}else if(a-d==0 && b-d==0 && c-d==0 ){

return c;

}

}

int main()

{

int a,b,c,d;

printf("Introdu A= ");

scanf("%d",&a);

printf("Introdu B= ");

scanf("%d",&b);

printf("Introdu C= ");

scanf("%d",&c);

printf("Introdu D= ");

scanf("%d",&d);

printf("MAx este %d \n",max\_elm(a,b,c,d));

printf("MIN este %d\n",min\_elm(a,b,c,d));

printf("Divizor comun %d\n ",divivor(a,b,c,d));

printf("multiplu\_min\_comun este %d\n",multiplu\_min(a,b,c,d));

printf("e\_divizor este %d\n",e\_divizor(a,b,c,d));

printf("f\_multiplu este %d\n",f\_multiplu(a,b,c,d));

return 0;

}

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.#include <stdio.h>

#include <stdlib.h>

#include <limits.h>

int suma(int a){

int sum=0;

while(a!=0){

sum=sum+a%10;

a=a/10;

}

return sum;

}

int max\_n(int a){

int max,max\_aux=INT\_MIN;

while(a!=0){

if(max\_aux<max){

max\_aux=max;

}

max=a%10;

a=a/10;

}

return max\_aux;

}

int divizori(int a){

int count=0;

for(int i=1;i<=a;i++){

if(a%i==0){

count++;

}

}

return count;

}

int divizori\_sum(int a){

int sum=0;

for(int i=1;i<=a;i++){

if(a%i==0 && i<a){

sum+=i;

}

}

return sum;

}

const char \*n\_perfecte(int a){

int sum=0;

for(int i=2;i<=a/2;i++){

if(a%i==0 ){

sum+=i;

}

}

if (a==sum){

return "Este numar perfect";

}else{

return "Nu este numar perfect";

}

}

const char \*prim(int a){

int nr=0;

for(int i=2;i<=a/2;i++){

if(a%i==0 ){

nr++;

}

}

if (nr==0){

return "Este numar prim";

}else{

return "Nu este numar prim";

}

}

int main()

{

int a,b,c,d;

printf("Introdu A= ");

scanf("%d",&a);

printf("Introdu B= ");

scanf("%d",&b);

printf("Introdu C= ");

scanf("%d",&c);

printf("Introdu D= ");

scanf("%d",&d);

printf("% Suma elementului %d este = %d / Suma elementului %d este = %d / Suma elementului %d este = %d / Suma elementului %d este = %d \n",a,suma(a),b,suma(b),c,suma(c),d,suma(d));

printf("MAx din %d este= / MAx din %d este= / MAx din %d este= /MAx din %d este= \n ",a,max\_n(a),b,max\_n(b),c,max\_n(c),d,max\_n(d));

printf(" %d are %d divizori / %d are %d divizori / %d are %d divizori / %d are %d divizori \n ",a,divizori(a),b,divizori(b),c,divizori(c),d,divizori(d));

printf(" numarul %d are suma divizorilor = %d / numarul %d are suma divizorilor = %d / numarul %d are suma divizorilor = %d / numarul %d are suma divizorilor = %d \n ",a,divizori\_sum(a),b,divizori\_sum(b),c,divizori\_sum(c),d,divizori\_sum(d));

printf("numarul %d %s",a,n\_perfecte(a),b,n\_perfecte(b),c,n\_perfecte(c),d,n\_perfecte(d));

printf("numarul %d %s",a,prim(a),b,prim(b),c,prim(c),d,prim(d));

return 0;

}

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 <stdio.h>

#include <stdlib.h>

int max\_palindrom(int a,int b,int c,int d){

int sum\_a=0,sum\_b=0,sum\_c=0,sum\_d=0;

while(a>0){

int r\_a=a%10;

sum\_a=(sum\_a\*10)+r\_a;

a=a/10;

}

while(b>0){

int r\_b=b%10;

sum\_b=(sum\_b\*10)+r\_b;

b=b/10;

}

while(c>0){

int r\_c=c%10;

sum\_c=(sum\_c\*10)+r\_c;

c=c/10;

}

while(d>0){

int r\_d=d%10;

sum\_d=(sum\_d\*10)+r\_d;

d=d/10;

}

if(sum\_a>sum\_b && sum\_a>sum\_c && sum\_a>sum\_d){

return sum\_a;

}else if(sum\_b>sum\_a && sum\_b>sum\_c && sum\_b>sum\_d){

return sum\_b;

}else if(sum\_c>sum\_a && sum\_c>sum\_b && sum\_c>sum\_d){

return sum\_c;

}else if(sum\_d>sum\_a && sum\_d>sum\_b && sum\_d>sum\_c){

return sum\_d;

}

}

int min\_palindrom(int a,int b,int c,int d){

int sum\_a=0,sum\_b=0,sum\_c=0,sum\_d=0;

while(a>0){

int r\_a=a%10;

sum\_a=(sum\_a\*10)+r\_a;

a=a/10;

}

while(b>0){

int r\_b=b%10;

sum\_b=(sum\_b\*10)+r\_b;

b=b/10;

}

while(c>0){

int r\_c=c%10;

sum\_c=(sum\_c\*10)+r\_c;

c=c/10;

}

while(d>0){

int r\_d=d%10;

sum\_d=(sum\_d\*10)+r\_d;

d=d/10;

}

if(sum\_a<sum\_b && sum\_a<sum\_c && sum\_a<sum\_d){

return sum\_a;

}else if(sum\_b<sum\_a && sum\_b<sum\_c && sum\_b<sum\_d){

return sum\_b;

}else if(sum\_c<sum\_a && sum\_c<sum\_b && sum\_c<sum\_d){

return sum\_c;

}else if(sum\_d<sum\_a && sum\_d<sum\_b && sum\_d<sum\_c){

return sum\_d;

}

}

int max\_comb(int a){

int v[100],i1=0,aux,num=0;

while(a>0){

int r\_a=a%10;

v[i1]=r\_a;

i1++;

a=a/10;

}

for(int i=0;i<i1;i++){

for(int k=i+1;k<i1;k++){

if(v[i]<v[k]){

aux=v[i];

v[i]=v[k];

v[k]=aux;

}

}

}

for(int i=0;i<i1;i++){

num=(num\*10)+v[i];

}

return num;

}

int min\_comb(int a){

int v[100],aux,i1=0,num=0;

while(a>0){

int r\_a=a%10;

v[i1]=r\_a;

i1++;

a=a/10;

}

for(int i=0;i<i1;i++){

for(int k=i+1;k<i1;k++){

if(v[i]>v[k]){

aux=v[i];

v[i]=v[k];

v[k]=aux;

}

}

}

for(int i=0;i<i1;i++){

if(v[i]==0){

for (int i = i1; i >= 2; i--)

v[i + 1] = v[i];

i1 = i1 + 1;

v[2] = 0; break;

}

}

//printf("\n i1= %d",i1);

for(int i=0;i<i1;i++){

// printf("%d ",v[i]);

}

for(int i=0;i<i1;i++){

num=(num\*10)+v[i];

}

return num;

}

int main(){

int a,b,c,d;

printf("Introdu A= ");

scanf("%d",&a);

printf("Introdu B= ");

scanf("%d",&b);

printf("Introdu C= ");

scanf("%d",&c);

printf("Introdu D= ");

scanf("%d",&d);

printf("\Cel mai mare dintre oglinditul numerelor date %d \n",max\_palindrom(a,b,c,d));

printf("\Cel mai mic dintre oglinditul numerelor date %d \n",min\_palindrom(a,b,c,d));

printf("A(%d). Cel mai mare format dintre cifrele fiecarui numar respectiv %d \n",a,max\_comb(a));

printf("B(%d). Cel mai mare format dintre cifrele fiecarui numar respectiv %d \n",b,max\_comb(b));

printf("C(%d). Cel mai mare format dintre cifrele fiecarui numar respectiv %d \n",c,max\_comb(c));

printf("D(%d). Cel mai mare format dintre cifrele fiecarui numar respectiv %d \n",d,max\_comb(d));

printf("A(%d). Cel mai mic format dintre cifrele fiecarui numar respectiv %d \n",a,min\_comb(a));

printf("B(%d). Cel mai mic format dintre cifrele fiecarui numar respectiv %d \n",b,min\_comb(b));

printf("C(%d). Cel mai mic format dintre cifrele fiecarui numar respectiv %d \n",c,min\_comb(c));

printf("D(%d). Cel mai mic format dintre cifrele fiecarui numar respectiv %d \n",d,min\_comb(d));

return 0;

}