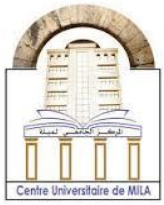
University Center Abdelhafid Boussouf -Mila-
Institute of Science and Technology

Computer Science 2



2023/2024

Practical work 1: functions in C

Part I

Exercise 1:

Create a function in C without parameters and without a return value to find the sum, product and division of two numbers entered by the user

Solution

```
#include <stdio.h>
void compute()
{
   double x,y,z,a,b ;
   printf("Enter a:\n");
   scanf("%lf",&a);
   printf("Enter b:\n");
   scanf("%lf",&b);
   x=a+b;
   y=a*b;
   z=a/b;
   printf("%lf+%lf=%lf\n",a,b,x);
   printf("%lf*%lf=%lf\n",a,b,y);
   printf("%lf/%lf=%lf\n",a,b,z);

}
int main() {

   compute();

   return 0;
}
```

Exercise 2:

Write a function sub-program in C to find the rank of every digit in a given number, and then displays the total number of digits.

Example:
The given number: 1356
The rank of: 6=0, 5=1, 3=2, 1=3.
The total number of digits=4

Solution

```
#include <stdio.h>
void Rank(long a)
{
   printf("The rank of every digits in the number %d:\n",a);
  int i=0,b;
  while(a!=0)
  {
    b=a%10;
    printf("[%d]=%d\n",b,i);
    i++;
    a=a/10;

  }
  printf("The total number of digits=%d\n----------\n",i);
}
int main() {

   Rank(177235);
   Rank(1289651);

   return 0;
}
```

Exercise 3:

1- Write a C program to find divisors of an integer using functions.
2- Use the previous function to create another function that finds all of divisors for a specified interval.

Solution

Function program to find divisors

```
#include <stdio.h>
 void divisors(int a)
   {int i=1;
   printf("divisors of %d:\n",a);
   while(i<=a)
   {
     if(a%i==0)
   {
      printf("%d\n",i);
    }
    i++;
   }


   }

int main() {

divisors(10);

   return 0;
}
```

Divisors in interval using functions

```
#include <stdio.h>
 void divisors(int a)
   {int i=1;
   printf("divisors of %d:\n",a);
   while(i<=a)
   {
     if(a%i==0)
   {
      printf("%d\n",i);
    }
    i++;
   }
   printf("----------------------\n");

   }
   void divb(int c,int d)
   {

      int j;
      for(j=c; j<=d;j++)
      {
         divisors(j);
      }

   }

int main() {
```

	divb(1,10);
	
	return 0;
	}

Part II: Recursive function

Exercise 4:

Write a function to find the sum of the square of the numbers from 1 to a.
The mathematical expression of the function is given as

$$f(a) = a^2 + (a-1)^2 + (a-2)^2 + \cdots + 1$$

Example: $f(5) = 5^2 + 4^2 + 3^2 + 2^2 + 1 = 55$

Solution

```
#include <stdio.h>

int sumsquar(int a)
{
   if(a==0)
   {
      return 0;
   }
   else
   {
      return (a*a)+sumsquar(a-1);
   }
}

int main() {

  int b;
  b=sumsquar(5);
```

```
  printf("sumsquare=%d",b);
    return 0;
}
```

Exercise 5:

Write in C, a program of two functions, one to find the sum of odd numbers from **1** to **n** and the other to find the sum of even numbers from **1** to **m**.

Solution

Program	Output

#include <stdio.h>
 	
int odd(int n)
{
if(n==1)
{
return 1; //Base case
}
else
{
if(n%2!=0)
{
return n+odd(n-2);//recursive case
}
else
{
n=n-1;
return n+odd(n-2);//recursive case
}
}
}
int even(int m)
{
if(m==2) // Base case
{
return 2;
}
	sumodd=16
	sumeven=12

```
   else
   {
        if(m%2==0)
        {
        return m+even(m-2); //recursive case
        }
        else
        {
        m=m-1;
       return m+even(m-2); //recursive case
        }
   }
}

int main() {

  int sumodd,sumeven;
  sumodd=odd(7);
  sumeven=even(7);


printf("sumodd=%d\nsumeven=%d",sumodd,sumeven);

   return 0;
}
```

Proposed exercise as Homework:

Create a function in C to insert a float number between two closest integers without using **ciel( )** and **floor( )** functions.

Example: 10<11.62<12.