Larbi Ben M'hidi -Oum El Bouaghi- University
Faculty of Exact Sciences and Natural and Life Sciences
Mathematics and Computer Science Department

Academic year: 2023-2024

Level: 1st year "Computer Science & Mathematics"

Module: Algorithmic and Data Structures 2

TP n°2 "Solution"

Pedagogic objectives

→ Handle procedures & functions in C;

Exercise n°1

1.

```
#include<stdio.h>
int power( int x, int y)
{
int result =1,i;
for( i =0;i< y;i ++)
result = result *x;
return result ;
}
int main()
{
int x=2 ,y =3,r;
r=power( x ,y );
printf( "%d power %d = %d\n", x,y,r );}
```

2.

```
#include <stdio.h>
void power(int x,int y,int*r)
{
int i ;
*r=1;
for( i =0;i< y;i ++)
*r = *r*x;
}
int main()
{
int x=4 ,y =5,r;
power( x ,y ,&r );
printf ("%d power %d = %d\n" , x,y,r );
}
```

3.

```
#include <stdio.h>
void division( int a, int b, int * q, int * r)
{
*r = a % b;
*q = a / b;
}
int main()
{
int a=17, b=3, q, r;
division( a,b,&q,&r);
printf ( "%d=%d*%d+%d\n", a,b,q,r );
}
```

Exercise n°2

```
#include <stdio.h>
/*function prototypes*/
void grab(int [],int);
```

```
void display(int [],int);
float calculate_average (int[],int);
void find_min_max ( int [], int , int *, int * );
main()
{
int nb_val ;
int min, max, table[100];
float average;
printf ( "Number of elements:"); scanf ("%d", & nb_val);
grab(table, nb_val);
display( table, nb_val );
average = calculate_average ( table , nb_val );
printf ( "Average = %f\n", average);
find_min_max(table, nb_val , &min, &max );
printf ("Min = %d Max = %d\n", min, max );
}
/*  entry of array elements */
void grab(int tab[], int nb)
{
int i ;
printf ( "\n");
for ( i =0; i < nb ; i ++)
{
printf ("Value of tab[%d] = ", i);
scanf ("%d", &tab[i] );
}
}
/* Displaying array elements */
void display ( int tab[], int nb )
{
int i ;
printf ( "\n");
for ( i =0; i < nb ; i ++)
{
```

```
printf ("tab[%d] = %d\n", i , tab[ i ]);
}
printf ( "\n");
}
/* Calculating the average */
float calculate_average ( int tab[ ], int nb)
{
float average;
int sum;
int i;
sum = 0;
for ( i =0; i < nb ; i ++)
{
sum = sum + tab[ i ];
}
average = sum /( double (nb));
return average;
}
/* the min and max of the  array */
void find_min_max ( int tab [], int nb, int *pmin , int *pmax )
{
int val_min , val_max ;
int i ;
val_min = tab[0];
val_max = tab[0];
for ( i =0; i < nb ; i ++)
{
if (tab[i] < val_min)
{
val_min = tab[i];
}
else if (tab[i] > val_max )
{
val_max = tab[i];
```

```
}
}
*pmin = val_min ;
*pmax = val_max ;
}
```

Exercise n°3

```
#include <stdio.h>
/* Prototypes of functions n */
void ACQUIRE( int *);
float F( float);
void READ_VECTOR (float[], int N);
void CALCULATE_VALUES(float [], float [], int N);
void DISPLAY_TABLE ( float [], float [], int N);
main()
{
/* Declaration of global variables */
float X[100]; /* values of X */
float V[100]; /* values of F(X) */
int Nb;
/* Function calls */
ACQUIRE(&Nb); /* 1 <= Number <= 100 */
READ_VECTOR(X, Nb);
CALCULATE_VALUES(X, V, Nb);
DISPLAY_TABLE(X, V, Nb);
}

/* Definition of the ACQUIRE function */
void ACQUIRE( int *N)
{
do
{
printf ( "Enter an integer between 1 and 100: ");
scanf ("%d",N);
```

```
 } while ((*N<1)||(*N>100));
}
/* Definition of the READ_VECTOR function */
void READ_VECTOR ( float T[], int N)
{
/* Fills an array T of order N with real numbers entered from
the keyboard */
/* Declaration of local variables */
int I;
/* Fill the array */
printf ( "Enter %d real numbers:\n", N);
for (I=0; I<N; I++)
 scanf ( "%f", &T[I]);
}
/* Definition of function F */
float F( float X)
{
/* Returns the numerical value of the polynomial defined by F(X)
= X^3-2X+1 */
return (X*X*X - 2*X + 1);
}
/* Definition of the CALCULATE_VALUES function */
void CALCULATE_VALUES( float X[], float V[], int N)
{
/* Declaration of local variables */
int I;
for (I=0; I<N; I++)
 {
V[ I] = F(X[I]);
}}
/* Definition of the DISPLAY_TABLE function */
void DISPLAY_TABLE(float X[], float V[], int N)
{
/* Declaration of local variables */
```

```
int I;
/* Show table */
printf ( "\nX: ");
for (I=0; I<N; I++)
printf ( "%f", X[I]);
printf ( "\nF(X): ");
for (I=0; I<N; I++)
printf ( "%f", V[I]);
printf ( "\n");}
```