

Chapter 1: Subprograms (Functions & Procedures)

Correction of exercise series

TD

1. Write a subroutine that displays the multiples of an integer n that are between two limits a and b ; test this procedure in the main function

```
#include<stdio.h>
void DisplayMultiple(int n ,int a,int b)
{int i ;
for(i=a/n; i*n<=b;i++)
printf("%d\t",i*n);
}
int main()
{ int n,a,b;
printf("Enter n :\n");
scanf("%d",&n);
printf("Entre limits a & b :\n");
scanf("%d%d",&a,&b);
DisplayMultiple(n,a,b);
return 0;
}
```

2. Write a C program that finds the max of four integers using a function **Max** that finds the max of two integers.

```
#include<stdio.h>
int Max(int a,int b)
{ if(a>b) return a;
return b;
}
int main()
{ int a,b,c,d,max;
printf("Entrez quatre entiers:\n");
scanf("%d%d%d%d",&a,&b,&c,&d);
max=Max(Max(a,b),Max(c,d));
printf("les max est: %d\n",max);
return 0;
}
```

3. Create a function that returns the number of letters in a character string passed as a parameter. Its prototype will be: *int numberLet(char text[]);* .

```
int numberLet (char text[]) {
int nb = 0,i=0;
while (text[i] != '\0'){
if((text[i]>='a' && text[i]<='z') || (text[i]>='A' && text[i]<='Z'))
nb++;
i++; }
return nb;
}
int main() {
char T[30];
```

```

printf("enter a string character : ");
gets ( T);
printf("\ number of letters in this string %d letters ", numberLet (T));
return 0;
}

```

4. Write a function that searches if a value is present in an array (integers). It will return (-1) if the value is not found and the position of the value in the array otherwise.

```

#include<stdio.h>
int findVal(int x,int t[],int n)
{ int i;
  for (i=0;i<n;i++)
    if (t[i]==x) return i ;
return -1;
}

int main()
{ int A[20], size , val ,i;
  printf("Enter size of the vector:\n");
  scanf("%d",&size);
  printf("enter elements of vector ");
  for (i=0;i<size;i++)
scanf("%d",&A[i]);
printf("Enter the value you want to find :\n");
scanf("%d",&val);
if (findVal(val,A,size) ==-1)
  printf (" this value does not exist in the vector");
else
  printf (" this value exist in %d th position",findVal(val,A,size));
}

```

5. The following program calculates the number of digits of an integer; he uses a procedure for this reason.

```

#include<stdio.h>
void myproc(int n , int s)
{ int s=0;
  while(n!=0) {
    s=s+1;
    n=n / 10;}
}
int main ( ){
int a, x=0;
printf ( " enter a positive integer: " );
scanf("%d",&a);
myproc(a,x);
printf (" the number of digits of %d is : %d ", a,x); }

```

- Execute the program and detect the error then correct it.

enter a positive integer: 2548

the number of digits of 2548 is :0

error : the second parameter will be passed by reference

```

void myproc(int n , int s)--->void myproc(int n , int * s)
s=s+1;--->*s=*s+1;
myproc(a,x); ---> myproc(a,&x);

```

- Replace the procedure with a function

```

#include<stdio.h> int main ( ){
int myproc(int n ) int a;

```

```

{   int s=0;           printf( " enter a positive integer: " );
    while(n!=0) {     scanf("%d",&a);
        s=s+1;       printf(" the number of digits of  %d is : %d ", a, myproc(a)); }
        n=n / 10;}
    return s;  }

```

6. Either the following function:

```

void conv (int n) {
    if (n<2)
        printf ("%d",n);
    Else  {conv (n/2) ;
           printf ("%d",n%2); } }

```

•.

• Calculate conv(23)? The procedure displays : 10111

• What does this function do? **Convert to binary code a decimal value**

• Test this function in a main function.

```

int main ()
{ int n =52;
  conv(n);
}

```

Generalize this function to do other similar roles : convert a decimal to b base (b<10)

```

void conv (int n, int b ) {
    if (n<b)
        printf ("%d",n);
    Else  {conv (n/b) ;
           printf ("%d",n%b); } }

```

7. Write a recursive function *Nb_div* which calculates the number of divisors of a positive integer
Write a function *Is_prime* which uses the *Nb_div* function to check if a positive integer is prime or not
Write the *main()* function which tests the *Is_prime* function

```

#include<stdio.h>
int nbdiv (int n,int b )
{ if (n==b) return 1;
  else if (n%b==0)
    return 1+nbdiv(n,b+1);
  else
    return nbdiv (n,b+1) ;
}
int isPrime(int n )
{if (nbdiv(n,1)==2) return 1;
 else return 0;
 }
int main(){
  int n,b ;
  printf ("n= ");
  scanf("%d",&n) ;
  if (isPrime(n))
    printf ("%d is prime ",n);
  else
    printf ("%d is not prime ",n);
}

```

}

TP

1. Write a subroutine which displays a number of seconds in minutes and seconds; test this subroutine in the main function.

```
int dispTime (int s )
{ int sec,min;
  sec=s%60;
  min=s/60;
  printf ("%dsec=%dmin:%dsec",s,min,sec);
}
```

```
int main(){ int t;
  printf ("seconds= ");
  scanf("%d",&t) ;
  dispTime(t);
}
```

2. Write a function that returns the integer part of a real number; use this function to check if the value of an entered real is integer or not.

```
int intPart (float x)
{ int i;
  for (i=1;i<=x;i++);
  return i-1 ;
}
```

```
int main(){
  float x;
  printf ("enter a real :");
  scanf("%f",&x) ;
  printf( " %d \n",intPart(x));
  if(intPart(x)==x)
    printf ("value of %f is an integer",x);
  else
    printf ("value of %f is an real",x);
}
```

3. write a procedure that returns the average of two real numbers using the following approaches:

- Utilizing a **global** variable
- Employing “**variable passing**”
- Rewrite this procedure as a **function**

```
min ( float x, float y) {
  if (x <= y)
    printf( "%f \n " , x);
  else
    printf( "%f \n " , y);
}
```

```

float m;
min ( float x, float y) {
    if (x <= y)
        m = x;
    else
        m = y;
}

```

```

min ( float x, float y, float *m) {
    if (x <= y)
        *m = x;
    else
        *m = y;
}

```

```

float min ( float x, float y) {
    if (x <= y)
        return x;
    return y;
}

```

4. Write a subroutine **sum_tab** that calculates the sum of the elements of a vector. Write a program that reads two vectors T1 and T2 and calculate their sum together.

```

#include <stdio.h>
float sumTab (float t[],int n){
    int i ; float s=0;
    for (i=0 ;i<n ;i++)
        s=s+t[i];
    return s;
}
int main(){
    float tab1[100] , tab2[100],sum;
    int N1,N2,i ;
    printf("enter the number of elements of tab1<=100");
    scanf("%d", &N1);
    for ( i=0;i<N1;i++){
        printf("%d=>" , i);
        scanf("%f", &tab1[i]);
    }
    printf("enter the number of elements of tab2<=100");
    scanf("%d", &N2);
    for ( i=0;i<N2;i++){
        printf("%d=>" , i);
        scanf("%f", &tab2[i]);
    }
    sum= sumTab (tab1,N1)+ sumTab (tab2,N2);
    Printf(" sum of 2 vectors : %.2f",sum)
}

```

5. Write a recursive function that calculates the GCD of two positive integers.

```

#include<stdio.h>
int GCD (int a,int b )
{ if (b==0) return a;
else
    return GCD(b,a%b);
}

```

```

int main(){
int a,b ;
printf ("enter 2 integers : ");
scanf("%d%d",&a,&b);
printf ("the GCD of %d and %d is %d ",GCD(a,b));
}

```

6. Write the function *Puis(x:real, n: integer)* which calculates X^n . use this function in a main function to calculate X^n for any value of X and n (write the function using two methods (iterative and recursive).

```

#include<stdio.h>
float puis (float x,int n )
{ float p=1;
  int i;
  for(i=1;i<=n;i++)
  p=p*x ;
  return p;
}
int main(){
  float X;int n ;
  printf ("enter a real : ");
  scanf("%f",&X);
  printf ("enter an integer:");
  scanf("%d",&n);
  if((n==0)&& (X==0))
    printf("non-appointment status");
  else if(n>=0)
    printf (".2f^%d=.2f",X,n,puis(X,n));
  else
    printf (".2f^%d=.2f",X,n,1/puis(X,-n));
}

```