CHAPTER 8 - FUNCTION

I. EXERCISES WITH SOLUTION

Exercise 1: Write a C program uses functions to calculate area and perimeter of a circle. Input radius from the keyboard then calculate area and perimeter of given circle using functions have written.

- **Solution**
- **Pseudo code**

BEGIN

INPUT Radius

CALL calcArea function with parameter radius

CALL calcPerimeter with parameter radius

END

MODULE calcArea with argument radius

Area =M_PI*radius*radius

RETURN Area

END_MODULE

MODULE calcPerimeter with argument radius

Perimeter =2*M_PI*radius

RETURN Perimeter

END_MODULE

- **C code**

```
/*Write function to calculate area and perimeter of circle.
Input radius from the key board. Calculate area and perimeter
uses function.
date writen:09.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
```

```
#include<math.h>
//function prototype
float calcArea(float radius);
float calcPerimeter(float radius);
void main(void)
{
     //declare variable
     float radius;
     float area;
     float perimeter;
     //Clear screen
     clrscr();
     //input radius
     printf("\nEnter the radius of circle please:");
     scanf("%f",&radius);
     //call function
     area=calcArea(radius);
     perimeter=calcPerimeter(radius);
     //display results to the screen
     printf("\nArea of circle is:%.2f",area);
     printf("\nPerimeter of circle is:%.2f",perimeter);
     printf("\nPress any key to continue");
     getch();//stop screen to view result
}
//Define functions
float calcArea(float radius)
{
     float area;
     area=M_PI*radius*radius;
     return area;
}
float calcPerimeter(float radius)
```

```
{
    float perimeter;
    perimeter=2*M_PI*radius;
    return perimeter;
}
```

Exercise 2: Write a C program solves all tasks below:

1. Write a function to input gross pay and tax rate
2. Write a function to calculate tax has two arguments gross pay, tax rate and return tax as formula tax=gross pay * tax rate.
3. Write a function has three arguments gross pay, tax rate, tax. Display them to the screen.
4. Using functions you have written in 1- 3 to input gross pay, tax rate from keyboard then calculate tax and display them to the screen.

- **Solution**
- **Pseudo code**

```
BEGIN
    CALL inputGrossPayAndTaxRate with parameter grossPay and taxRate
    CALL calcTax with parameter grossPay and taxRate
    CALL displayResult with parameter grossPay, taxRate and tax
END
MODULE inputGrossPayAndTaxRate with arguments grossPay and taxRate
    INPUT grossPay
    INPUT taxRate
END_MODULE
MODULE calcTax with argument grossPay and taxRate
    tax=grossPay*taxRate/100.0
    RETURN tax
END_MODULE
MODULE displayResult with arguments grossPay, taxRate, tax
```

DISPLAY grossPay

DISPLAY taxRate

DISPLAY tax

END_MODULE

- **C code**

```
/*Write functions to input gross pay, tax rate, calculate tax and
display them to the screen. Input gross pay, tax rate from the key
board. Calculate tax, display them to the screen.
date writen:09.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
//function prototype
void inputGrossPayAndTaxRate(float *grossPay,int *taxRate);
float calcTax(float grossPay,int taxRate);
void displayResult(float grossPay,int taxRate,float tax);
void main(void)
{
      //declare variable
      float grossPay;
      float tax;
      int taxRate;
      //Clear screen
      clrscr();
      //call function inputGrossPayAndTaxRate to in put gross pay and tax rate
      inputGrossPayAndTaxRate(&grossPay,&taxRate);
      //call function calcTax to find out tax
      tax=calcTax(grossPay,taxRate);
      //call function displayResult to display gross pay, tax rate and tax
      displayResult(grossPay,taxRate,tax);
      printf("\nPress any key to continue");
      getch();//stop screen to view result
```

```
}
//Define functions
void inputGrossPayAndTaxRate(float *grossPay,int *taxRate)
{
      printf("\nEnter your gross pay please:");
      scanf("%f",grossPay);
      printf("\nEnter the tax rate please:");
      scanf("%d",taxRate);
}


float calcTax(float grossPay,int taxRate)
{
      float tax;
      tax=grossPay*taxRate/100.0;
      return tax;
}
void displayResult(float grossPay,int taxRate,float tax)
{
      printf("\nYour gross pay is:%.2f",grossPay);
      printf("\nThe tax rate is:%d%%",taxRate);
      printf("\nYour tax is:%.2f",tax);
}
```

Exercise 3: Write a C program uses function to create menu as below:

1. Enter the size and integer array from the keyboard.
2. Display array to the screen.
3. Find and display to the screen min even number and max odd number.
4. Count prime number in the array. Display result to the screen.
5. Calculate and display total square number in the array to the screen.
6. Sort array ascending and display it after sorting.
7. Exit from program

Enter your choice please:

When user chooses 1 then program permits user to input number elements of array and array from the keyboard. Choosing 2 then display given array to the screen, each element use 5 positions to display. Choosing 3 then find and display to the screen min even number and max odd number in the array. Choosing 4 then count prime number in the array. Display result to the screen. If no prime number in the array then display message like “no prime number in the array”. Choosing 5 then calculate and display total square number in the array to the screen. Choosing 6 then sort array ascending and display it after sorting. Choosing 7 then exit from program.

- **Solution**
- **Pseudo code**

```
BEGIN
    REPEAT
        CALL menu
        INPUT choice
        CASE choice OF
            Case 1: CALL input function
            Case 2: CALL display function
            Case 3: CALL findMinMax function
                DISPLAY result
            Case 4: CALL countPrime function
                DISPLAY result
            Case 5: CALL calcTotalSquareNumber function
                DISPLAY result
            Case 6: CALL sortArray and display function
        END_CASE
    UNTIL choice=7
END
```

```
MODULE inputArray with arguments n (pointer) and array a
    INPUT n
    FOR i=0 TO n-1 DO
            INPUT a[i]
    END_FOR
END_MODULE
MODULE displayArray with arguments n and array a
    FOR i=0 TO n-1 DO
            DISPLAY a[i]
    END_FOR
END_MODULE
MODULE findMinEvenMaxOdd with arguments min and max (pointer)
    min=32767
    max=-32768
    FOR i=0 TO n-1 DO
            IF a[i] MOD 2 = 0 AND a[i]<min THEN
                    min=a[i]
            END_IF
            IF a[i] MOD 2<>0 AND a[i]>max THEN
                    max=a[i]
            END_IF
    END_FOR
END_MODULE
MODULE checkPrime with argument num
    flag=0
    FOR i=2 TO num-1 DO
            IF num MOD i=0 THEN
                    flag=1
                    Exit from for loop
```

```
            END_IF
    END_FOR
    RETURN 1-flag {1 if num is prime and 0 if num is not prime}
END_MODULE
MODULE countPrime with arguments n and array a
    count=0;
    FOR i=0 TO n-1 DO
            CALL checkPrime with parameter a[i]
            IF result return from checkPrime is true (1) THEN
                    count = count + 1
            END_IF
    END_FOR
    RETURN count
END_MODULE
MODULE checkSquare with argument num
    flag=0
    IF num>=0 AND Sqrt(num)*Sqrt(num)=num THEN
            flag = 1
    END_IF
    RETURN flag
END_MODULE
MODULE calcTotalSquare with arguments n and array a
    total=0
    FOR i=0 TO n-1 DO
            CALL checkSquare with parameter a[i]
            IF result return from checkSquare is true (1) THEN
                    total = total + a[i]
            END_IF
    END_FOR
```

```
        RETURN total
    END_MODULE
    MODULE sortArray with arguments n and array a
        FOR i=0 TO n-2 DO
            FOR j=i TO n-1 DO
                IF a[j]<a[i] THEN
                    temp=a[i]
                    a[i]=a[j]
                    a[j]=temp
            END_FOR
        END_FOR
    END_MODULE
```

- **C code**

```
 /*Program uses menu and function.
date writen:09.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
//function prototype
void menu(void);
void inputArray(int a[],int *n);
void displayArray(int a[],int n);
void findMinEvenMaxOdd(int a[],int n,int *max,int *min);
int checkPrime(int num);
int countPrime(int a[],int n);
int checkSquare(int num);
int calcTotalSquare(int a[],int n);
void sortArray(int a[],int n);
```

```
void main(void)
{
     //declare variable
     int arr[100];
     int n;
     int count;
     int max;
     int min;
     int total;
     int choice;
     //Clear screen
     clrscr();
     do{
          menu();
          scanf("%d",&choice);
          switch(choice)
          {
               case 1:inputArray(arr,&n);
                    break;
               case 2:displayArray(arr,n);
                    break;
               case 3:findMinEvenMaxOdd(arr,n,&max,&min);
                      if(min==32767)
                    printf("\n There is not any even number in
the array");
                      else
                    printf("\n Min even in the array
is:%d",min);
                      if(max==-32768)
                    printf("\n There is not any odd number in
the array");
                      else
```

```
                    printf("\n Max odd in the array is:%d",max);
                      break;
               case 4:count=countPrime(arr,n);
                      if(count==0)
                    printf("\nThere is not any prime number in
the array");
                      else
                    printf("\nThere are %d prime numbers in the
array",count);
                      break;
               case 5:total=calcTotalSquare(arr,n);
                      if(total==0)
                    printf("\nThere is not any square number in
the array");
                      else
                    printf("\nTotal of all square numbers in the
array is:%d",total);
                      break;
               case 6: sortArray(arr,n);
                    displayArray(arr,n);

          }
     }while(choice!=7);
     printf("\nPress any key to continue");
     getch();//stop screen to view result

}
//Define functions
void menu(void)
{
     printf("\n\t1.Enter the size and integer array from the
keyboard");
```

```
     printf("\n\t2.Display the given array to the screen");
     printf("\n\t3.Find and display min even and max odd
number");
     printf("\n\t4.Count prime number in the array. Display the
result to the screen ");
     printf("\n\t5.Calculate and display total all square
numbers in the array");
     printf("\n\t6.Sort array ascending. Display array after
sorting to the screen");
     printf("\n\t7.Exit from program");
     printf("\n\tEnter your choice please:");

}
void inputArray(int a[],int *n)
{
     int i;//index of element
     printf("\nEnter number of elements in a array please:");
     scanf("%d",n);
     //input array
     for(i=0;i<*n;i++)
     {
          printf("\n Enter arr[%d]=",i+1);
          scanf("%d",&a[i]);
     }
}
void displayArray(int a[],int n)
{
     int i;//index of element
     printf("\nAll elements of the array are:");
     for(i=0;i<n;i++)
          printf("%5d",a[i]);
}
```

```
void findMinEvenMaxOdd(int a[],int n,int *max,int *min)
{
     int i;
     *max=-32768;//-32768 is min of integer data type
     *min=32767;//32767 is max of integer data type
     for(i=0;i<n;i++)
     {
          if(a[i]%2!=0 && a[i]>*max)
               *max=a[i];
          if(a[i]%2==0 && a[i]<*min)
               *min=a[i];

     }
}
int checkPrime(int num)
{
     int flag;
     int i;
     flag=0;
     for(i=2;i<num;i++)
          if(num%i==0)
          {
               flag=1;
               break;
          }
     return 1-flag;
}
int countPrime(int a[],int n)
{
     int count=0;
     int i;
     for(i=0;i<n;i++)
```

```
          if(checkPrime(a[i])==1)
               count++;
     return count;
}
int checkSquare(int num)
{
     int flag=0;
     if(num>=0&&(int)sqrt(num)*(int)sqrt(num)==num)
          flag=1;
     return flag;
}
int calcTotalSquare(int a[],int n)
{
     int total=0;
     int i;
     for(i=0;i<n;i++)
          if(checkSquare(a[i])==1)
               total+=a[i];
     return total;
}
void sortArray(int a[],int n)
{
     int i;
     int j;
     int temp;
     for(i=0;i<n-1;i++)
          for(j=i;j<n;j++)
               if(a[j]<a[i])
               {
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
```

```
                    }
}
```

Exercise 4: Write functions to perform following tasks:

1. Function to input matrix has n rows and m column
2. Function to display given matrix to the screen.
3. Function to sort all rows of matrix ascending.
4. Function to calculate total of all elements of matrix.

Write a C program uses functions you have written from 1 to 4 to input matrix has n rows and m columns from the keyboard. Display given matrix to the screen. Sort all rows of matrix ascending and display it after sorting. Calculate total of all elements of matrix and display result to the screen.

- Solution
- Pseudo code

```
BEGIN
    CALL inputMatrix
    CALL displayMatrix
    CALL sortMatrix
    CALL displayMatrix
    CALL calcTotal
    DISPLAY result
END
MODULE inputMatrix with argument matrix a, row n, column m
    INPUT n
    INPUT m
    FOR i=0 TO n-1 DO
        FOR j=0 TO m-1 DO
            INPUT a[i][j]
        END_FOR
    END_FOR
```

```
END_MODULE
MODULE outputMatrix with argument matrix a, row n, column m
    FOR i=0 TO n-1 DO
        FOR j=0 TO m-1 DO
            DISPLAY a[i][j]
        END_FOR
        Enter new line
    END_FOR
END_MODULE
MODULE sortMatrix with argument matrix a, row n, column m
    FOR i=0 TO n-1 DO
        FOR j=0 TO m-2 DO
            FOR k=j TO m-1 DO
                IF a[i][k]<a[i][j] THEN
                    temp=a[i][k]
                    a[i][k]=a[i][j]
                    a[i][j]=temp
                END_IF
            END_FOR
        END_FOR
    END_FOR
END_MODULE
MODULE calcTotal with argument matrix a, row n, column m
    total=0
    FOR i=0 TO n-1 DO
        FOR j=0 TO m-1 DO
            total=total+a[i][j]
        END_FOR
    END_FOR
```

RETURN total

END_MODULE

- C code

```
/*Program uses function.
date writen:09.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
//function prototype
void inputMatrix(int a[][10],int *n,int *m);
void displayMatrix(int a[][10],int n,int m);
void sortMatrix(int a[][10],int n,int m);
int calcTotal(int a[][10],int n,int m);
void main(void)
{
     //declare variable
     int arr[10][10];
     int n;
     int m;
     int total;
     //Clear screen
     clrscr();
     //call function inputMatrix
     inputMatrix(arr,&n,&m);
     //call function displayMatrix
     printf("\nAll elements of matrix\n");
     displayMatrix(arr,n,m);
     //call function sortMatrix
     sortMatrix(arr,n,m);
     //call function displayMatrix
```

```
     printf("\nAll elements of matrix after sorting\n");
     displayMatrix(arr,n,m);
     //call function calcTotal
     total=calcTotal(arr,n,m);
     printf("\nTotal of all elements is:%d",total);
     printf("\nPress any key to continue");
     getch();//stop screen to view result

}
//Define functions
void inputMatrix(int a[][10],int *n,int *m)
{
     int row;//row index
     int col;//column index
     printf("\nEnter number of rows please:");
     scanf("%d",n);
     printf("\nEnter number of columns please:");
     scanf("%d",m);
     for(row=0;row<*n;row++)
          for(col=0;col<*m;col++)
          {
               printf("arr[%d][%d]=",row+1,col+1);
               scanf("%d",&a[row][col]);
          }

}
void displayMatrix(int a[][10],int n,int m)
{
     int row;
     int col;
     for(row=0;row<n;row++)
     {
```

```
          for(col=0;col<m;col++)
               printf("%5d",a[row][col]);
          printf("\n");
     }
}
void sortMatrix(int a[][10],int n,int m)
{
     int i;
     int j;
     int k;
     int temp;
     for(i=0;i<n;i++)
          for(j=0;j<m-1;j++)
               for(k=j;k<m;k++)
                    if(a[i][k]<a[i][j])
                    {
                         temp=a[i][j];
                         a[i][j]=a[i][k];
                         a[i][k]=temp;
                    }
}
int calcTotal(int a[][10],int n,int m)
{
     int row;
     int col;
     int total=0;
     for(row=0;row<n;row++)
          for(col=0;col<m;col++)
               total+=a[row][col];
     return total;
}
```

II. EXERCISES WITHOUT SOLUTION

Exercise 1: Write functions perform tasks below:

1. Input length and width of rectangle
2. Calculate area of rectangle has length and width
3. Calculate perimeter of rectangle has length and width
4. Display length, width, area, perimeter of rectangle to the screen.

Using functions you have written from 1 to 4 to input length and width of rectangle. Calculate area and perimeter of rectangle. Display all information to the screen.

Exercise 2: Write functions perform tasks below:

1. Input three integers from the keyboard
2. Find min and max of three given integers.
3. Calculate minus of square max and square min
4. Find and display to the screen all divisors of three given integers.

Write a C program uses functions you have written from 1 to 4 to input three integers from the keyboard find and display to the screen min and max of three given integers, calculate and display to the screen minus of square max and square min, find and display to the screen all divisors of three given integers.

Exercise 3: Write functions to perform following tasks:

1. Write a function InputNumber(), which inputs a float number, validates the number to be between 0 and 100 inclusive and returns the number.
2. Write a function InputChoice() that inputs a character 'P' to output the number as a percentage or 'D' to output the number as a decimal value, validates the choice to be 'P' or 'D' and returns the Choice.
3. Write a function OutputResult that accepts the float number and character choice as input parameters, outputs the number as a percentage when choice is P or outputs the decimal number (divide the number by 100) when choice is D.

4. Using the functions you have written in 1-3, write a program to input a number (between 0-100), input a choice(P or D) and either output the number as a percentage or decimal number.

Known Results

Number	Choice	Expected Output
-1		Error message – number must be 0-100
50	X	Error message – type must be P or D
	P	50.00%
50	D	0.50

Exercise 4: What is output by the following program

a)

```
#include <stdio.h>
#include <conio.h>

void DoSomething(int Num2);


int main(void)
{
	int Num1 = 0;
	int Num2 = 3;
	printf("\nInside main: Num1 = %d Num2 = %d", Num1, Num2);
	DoSomething(Num2);
	printf("\nInside main: Num1 = %d Num2 = %d", Num1, Num2);
	return 0;
}

void DoSomething(int Num2)
{	int Num1 = 0;
	Num1 = Num1 + 5;
	printf("\nInside DoSomething: Num1 = %d Num2 = %d", Num1, Num2);
	return;
}
```

b)

```
#include <stdio.h>
#include <conio.h>

int DoSomething(int Num1);


int main(void)
{
      int Num1 = 4;
      int Num2 = 7;
      printf("\nInside main: Num1 = %d Num2 = %d", Num1, Num2);
      Num1 = DoSomething(Num1);
      printf("\nInside main: Num1 = %d Num2 = %d", Num1, Num2);
      return 0;
}

int DoSomething(int Num1)
{     int Num2 = 0;
      Num1 = Num1 + 2;
      printf("\nInside DoSomething: Num1 = %d Num2 = %d", Num1, Num2);
      return Num1;
}
```

Exercise 5: Write a C program uses function to create menu as below:

1. Enter the size and integer array from the keyboard.
2. Display array to the screen.
3. Find and display to the screen min positive and max negative number.
4. Count square number in the array. Display result to the screen.
5. Calculate and display to the screen total prime number in the array.
6. Sort array descending and display it after sorting.

7. Exit from program

Enter your choice please:

When user chooses 1 then program permits user to input number elements of array (>0 and <=50) and array from the keyboard. Choosing 2 then display given array to the screen, each element use 6 positions to display. Choosing 3 then find and display to the screen min positive number and max negative number in the array. Choosing 4 then count square number in the array. Display result to the screen. If no square number in the array then display message like “no square number in the array”. Choosing 5 then calculate and display total prime number in the array to the screen. Choosing 6 then sort array descending and display it after sorting. Choosing 7 then exit from program.

Exercise 6: Write functions perform following tasks:

1. Function to input matrix has n rows and m column.
2. Function to display given matrix to the screen.
3. Function to sort all columns of matrix ascending.
4. Function to calculate total of even elements of matrix.

Write a C program uses functions you have written in 1 - 4 to input matrix has n rows and m columns from the keyboard. Display given matrix to the screen. Sort all columns of matrix ascending and display it after sorting. Calculate total of all even elements of matrix and display result to the screen.

Hint: Reference exercise 4 in exercise with solution session.