CHAPTER 7 - POINTER

I. EXERCISES WITH SOLUTION

Exercise 1: Write a C program uses pointer to input number elements n (0<n<=20) and all elements of integer array from the keyboard. Enter integer m (m<=n). Display all elements of given array to the screen so that each line has m elements.

For example: n=10, a[0]=0, a[1]=1, a[2]=2, a[3]=3, a[4]=4, a[5]=5,a[6]=6, a[7]=7, a[8]=8, a[9]=9, m=4.

```
0       1       2       3
4       5       6       7
8       9
```

- Solution

- Pseudo code

```
BEGIN
        INPUT n
        {Validate n so that n>0 and n<=20}
        WHILE n<=0 OR n>20 DO
                DISPLAY ErrMess
                INPUT n
        END_WHILE
        {Input array}
        FOR i=0 TO n-1 DO
                INPUT a[i]
        END_FOR
        {Input and validate m so that m>0 and m<=n}
        INPUT m
        WHILE m<=0 OR m>n DO
                DISPLAY ErrMess
                INPUT m
```

```
    END_WHILE
    {Display to screen as requirement}
    FOR i=0 TO n-1 DO
        DISPLAY a[i]
        IF (i+1) MOD m = 0 THEN
            Carry return
        END_IF
    END_FOR
END
```

- C code

```c
/*Input array from keyboard. Display m elements in each line
date writen:08.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
void main(void)
{
    //declare variable
    int i;//index of array
    int n;//number of elements
    int m;//number of elements in each line
    int a[20];
    //Clear screen
    clrscr();
    //input and validate n
    printf("\nEnter the number of elements:");
    scanf("%d",&n);
    while(n<=0 || n>20)
    {
        printf("\nReenter n so that n>0 and n<=20:");
        scanf("%d",&n);
    }
    //Input the array
    for(i=0;i<n;i++)
    {
        printf("\nEnter the element %d:",i+1);
        scanf("%d",a+i);
    }
    //input and validate m
    printf("\nEnter the number of elements in each line:");
    scanf("%d",&m);
    while(m<=0 ||m>n)
    {
        printf("\nReenter m so that m>0 and m<=%d:",n);
        scanf("%d",&m);
    }
    //Display array to the screen as requirement
    printf("\n\n");
    for(i=0;i<n;i++)
    {
        printf("%5d",*(a+i));
        if((i+1)%m==0)
            printf("\n");
    }
    printf("\nPress any key to continue");
    getch();//stop screen to view result

}
```

Exercise 2: Write a C program uses dynamic allocation to input integer array from keyboard then do following tasks:

- Display given array to the screen.

- Display all square number in the array to the screen.
- Sort array descending and display it after sorting.
- **Solution**
- **Pseudo code**

```
BEGIN
    INPUT n
    Allocate memory to hold n integer elements.
    {Input the array}
    FOR i=0 TO n-1 DO
            INPUT a[i]
    END_FOR
    {Display the array}
    FOR i=0 TO n-1 DO
            DISPLAY a[i]
    END_FOR
    {Display all square number in the array}
    FOR i=0 TO n-1 DO
            IF a[i]=SQRT(a[i])*SQRT(a[i]) THEN
                    DISPLAY a[i]
            END_IF
    END_FOR
    {Sort the array and display it after sorting}
    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_IF
```

```
        END_FOR
    END_FOR
    FOR i=0 TO n-1 DO
        DISPLAY a[i]
    END_FOR
END
```

- **C code**

```
/*Dynamic allocate memory and input array from keyboard.
Display square numbers in the array. Sort descending and display
array after sorting
date writen:08.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main(void)
{
     //declare variable
     int i,j;//index of array
     int n;//number of elements
     int temp;
     int *a;
     //Clear screen
     clrscr();
     //input number of elements
     printf("\nEnter the number of elements:");
     scanf("%d",&n);
     //allocate memory
     a=(int *)calloc(n,sizeof(int));
     //Input the array
```

```
for(i=0;i<n;i++)
{
     printf("\nEnter the element %d:",i+1);
     scanf("%d",a+i);
}
//Display the array to the screen
printf("\nAll elements of the array are:");
for(i=0;i<n;i++)
     printf("%5d",*(a+i));

//Display all square numbers in the array
printf("\n\nAll square numbers in the array are:");
for(i=0;i<n;i++)
{
     if((int)sqrt(*(a+i))*(int)sqrt(*(a+i))==*(a+i))
          printf("%5d",*(a+i));
}
//sort and display
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;
          }
printf("\nAll elements of array after sorting:");
for(i=0;i<n;i++)
     printf("%5d",*(a+i));
free(a);
printf("\nPress any key to continue");
getch();//stop screen to view result
```

}

Exercise 3: Write a C program uses pointer to input integer array from the keyboard then do following tasks:

- Display the given array to the screen.
- Calculate average even and odd number in the array. Display result to the screen.
- Find max positive and min negative number in the array. Display result to the screen.
- **Solution**
- **Pseudo code**

```
BEGIN
    INPUT n
    Allocate memory to hold n integer
    {Input array}
    FOR i=0 TO n-1 DO
        INPUT a[i]
    END_FOR
    {Calculate average even and odd number in the array}
    countEven=0
    countOdd=0
    totalEven=0
    totalOdd=0
    FOR i=0 TO n-1 DO
        IF a[i] MOD 2=0 THEN
            countEven=countEven+1
            totalEven=totalEven+a[i]
        ELSE
            countOdd=countOdd+1
            totalOdd=totalOdd+a[i]
        END_IF
    END_FOR
    IF countEven>0 THEN
        averageEven =totalEven/countEven
        DISPLAY averageEven
    ELSE
        DISPLAY “No even number in the array”
    END_IF
    IF countOdd>0 THEN
        averageOdd =totalOdd/countOdd
        DISPLAY averageOdd

    ELSE
        DISPLAY “No odd number in the array”
    END_IF
    {Find max positive and min negative number in the array}
    max=0
    min=0
    FOR i=0 TO n-1 DO
        IF a[i]>0 AND a[i] >max THEN
            max=a[i]
        END_IF
        IF a[i]<0 AND a[i]<min THEN
            min=a[i]
        END_IF
    END_FOR
    DISPLAY max,min
END
```

- **C code**

```
/*Dynamic allocate memory and input array from keyboard.
```

```
Calculate average even and odd numbers in the array
Find max positive and min negative number in the array
date writen:08.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
void main(void)
{
    //declare variable
    int i;//index of array
    int n;//number of elements
    int countEven=0;
    int countOdd=0;
    int totalEven=0;
    int totalOdd=0;
    int min=0;
    int max=0;
    float averageEven;
    float averageOdd;
    int *a;//store beginning address of memory area
    //Clear screen
    clrscr();
    //input number of elements
    printf("\nEnter the number of elements:");
    scanf("%d",&n);
    //allocate memory
    a=(int *)calloc(n,sizeof(int));
    //Input the array
    for(i=0;i<n;i++)
    {
        printf("\nEnter the element %d:",i+1);
        scanf("%d",a+i);
    }
    //Display the array to the screen
    printf("\nAll elements of the array are:");
    for(i=0;i<n;i++)
        printf("%5d",*(a+i));

    //calculate average even and odd number in the array
    for(i=0;i<n;i++)
    {
        if(*(a+i)%2==0)
        {
            countEven++;
            totalEven+=*(a+i);
        }
        else
        {
            countOdd++;
            totalOdd+=*(a+i);
        }
    }
    if(countEven>0)
    {
        averageEven=(float)totalEven/countEven;
        printf("\nAverage even number in the
array:%.2f",averageEven);
    }
    else
        printf("\nNo even number in the array");
    if(countOdd>0)
    {
        averageOdd=(float)totalOdd/countOdd;
```

```
          printf("\nAverage odd number in the 
array:%.2f",averageOdd);
     }
     else
          printf("\nNo odd number in the array");

     //find min negative and max positive number in the array
     for(i=0;i<n;i++)
     {
          if(*(a+i)>0 && *(a+i)>max)
               max=*(a+i);
          if(*(a+i)<0 && *(a+i)<min)
               min=*(a+i);

     }
     printf("\nMin negative number in the array is:%d",min);
     printf("\nMax positive number in the array is:%d",max);
     printf("\nPress any key to continue");
     getch();//stop screen to view result
}
```

Exercise 4: Write a C program uses pointer to input matrix has n rows and m columns from the keyboard. Find transpose matrix of given matrix. Display two matrixes to the screen.

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

```
BEGIN
    {Input number of rows and columns of matrix}
    INPUT n
    INPUT m
    {Input matrix}
```

```
FOR row=0 TO n-1 DO
    FOR col=0 TO m-1 DO
        INPUT a[row][col]
    END_FOR
END_FOR
{Find transpose of given matrix}
FOR row=0 TO n-1 DO
    FOR col=0 TO m-1 DO
        b[col][row]=a[row][col]
    END_FOR
END_FOR
{Display two matrixes to the screen}
FOR row=0 TO n-1 DO
    FOR col=0 TO m-1 DO
        DISPLAY a[row][col], b[row][col]
    END_FOR
    Enter new line
END_FOR

END
```

- **C code**

```
/*Input matrix and find transpose of it. Display two matrixes to
the screen
date writen:08.07.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
void main(void)
```

```
{
     //declare variable
     int row;//row index of matrix
     int col;//column index of matrix
     int n;//number of rows
     int m;//number of columns
     int (*a)[10];
     int (*b)[10];//transpose matrix
     //Clear screen
     clrscr();
     //input number of rows and columns
     printf("\nEnter the number of rows:");
     scanf("%d",&n);
     printf("\nEnter the number of colunm:");
     scanf("%d",&m);
     //input matrix from the keyboard
     for(row=0;row<n;row++)
          for(col=0;col<m;col++)
          {
               printf("\na[%d][%d]=",row+1,col+1);
                        scanf("%d",*(a+row)+col);
          }
     //find transpose matrix
     for(row=0;row<n;row++)
          for(col=0;col<m;col++)
               *(*(b+col)+row)=*(*(a+row)+col);
     //Display two matrixes to the screen
     printf("\nAll elements of matrix a are:\n");
     for(row=0;row<n;row++)
     {
          for(col=0;col<m;col++)
               printf("%5d",*(*(a+row)+col));
          printf("\n");
     }
     printf("\nAll elements of transpose matrix  are:\n");
     for(row=0;row<n;row++)
     {
          for(col=0;col<m;col++)
               printf("%5d",*(*(b+row)+col));
          printf("\n");
     }
     printf("\nPress any key to continue");
     getch();//stop screen to view result
}
```

II. EXERCISES WITHOUT SOLUTION

Exercise 1: Write a C program uses pointer and dynamic allocate memory to input integer array from the keyboard. Then do following tasks:

- Display the given array to the screen.
- Sort array ascending. Display array after sorting to the screen.
- Display all prime numbers in the array to the screen.

Exercise 2: Write a C program uses pointer and dynamic allocate memory to input float array from the keyboard. Then do following tasks:

- Display the given array to the screen.
- Sort array descending. Display array after sorting to the screen.
- Calculate average positive and non positive number in the array.

Exercise 3: Write a C program uses pointer and dynamic allocate memory to input integer array from the keyboard. Then do following tasks:

- Display the given array to the screen.
- Calculate total all elements in the array. Display result to the screen.
- Display all square number in the array.

Exercise 4: Write a C program uses pointer and dynamic allocate memory to input integer array from keyboard. Then do following tasks;

- Display the given array to the screen.
- Count perfect number in the array. Display result to the screen.
- Find max even and min odd number in the array. Display result to the screen.

- Hint: To find max even and min odd number you should assign max even = -32768, min odd = 32767.

Exercise 5: Write a C program uses pointer to input matrix has n rows and m columns from the keyboard then do following tasks:

- Display given matrix to the screen.
- Calculate total of all element in even row. Display result to the screen.
- Sort all columns descending. Display matrix after sorting to the screen.

Exercise 6: Write a C program uses pointer to inputs exam score range from 1 to 100 of 40 students from the keyboard and store in the array. The program should count and print the number of outstanding scores (90-100), the number of high average scores (70-89), the number of satisfactory scores (50-69), and the number of unsatisfactory scores (1-49).