CHAPTER 5 - LOOP STRUCTURES

I. EXERCISES WITH SOLUTION

Exercise 1: Write a C program inputs positive integer n. Calculate and display to the screen total number from 1 to n.

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

```
BEGIN
    INPUT n
    total = 0
    FOR i = 1 TO n DO
        total = total + i
    END_FOR
    DISPLAY total
END
```

- **C code**

```
/*Program to input positive integer n. Calculate and display to
the screen total number from 1 to n.
date writen:27.06.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>
void main(void)
{
	//declare variable
```

```
    int n;
    int i;
    int total;
    //Clear screen
    clrscr();
    printf("\nEnter a positive integer please:");
    scanf("%d",&n);
    total=0;
    for(i=1;i<=n;i++)
         total=total+i;
    printf("\nTotal from 1 to %d is:%d",n,total);
    printf("\nPress any key to continue");
    getch();//stop screen to view result
}
```

Exercise 2: Write a C program inputs positive integer n. Display to the screen the message that n is prime number or not.

- Example screen result when program runs

```
Enter positive integer please:8
8 is not prime number
Enter positive integer please:7
7 is prime number
```

- Solution
- Pseudo code

```
  {n is prime number if it have only 1 and n divisor }
  BEGIN
      INPUT n
      flag = 0
      FOR i = 2 TO n-1 DO
```

```
            IF n MOD i = 0 THEN
                flag = 1{Mark n has addition divisor difference from 1 and n}
                EXIT from FOR loop
            END
        END_FOR
        IF flag = 1 THEN
            DISPLAY n + “ is not prime number”
        ELSE
            DISPLAY n + “ is prime number”
        END_IF
    END
```

- Solution

```
/*Program to input positive integer n. Display to the screen
n is prime number or not
date writen:28.06.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>

void main(void)
{
     //declare variable
     int n;
     int i;
```

```
	int flag;
	//Clear screen
	clrscr();
	printf("\nEnter a positive integer please:");
	scanf("%d",&n);
	flag=0;
	for(i=2;i<n;i++)
		if(n%i==0)
		{
			flag=1;
			break;
		}
	if(flag==1)
		printf("\n%d is not prime number",n);
	else
		printf("\n%d is prime number",n);

	printf("\nPress any key to continue");
	getch();//stop screen to view result
}
```

Exercise 3: Write a C program input integer n (n>0 AND n<=100). Display to the screen n is perfect number or not.

- Example screen result when program runs

```
Enter a positive integer: -10

Please reenter integer n so that n greater than zero and less than or equal 100: 101

Please reenter integer n so that n greater than zero and less than or equal 100: 20

20 is not perfect number
```

-Solution: Perfect number has sum of all divisors except it equals to it. Example 6 is perfect number because it has divisors is 1, 2, 3 and sum of divisors is 1 + 2 + 3 = 6.

- **Pseudo code:**

```
BEGIN
    INPUT n
    WHILE n<=0 OR n>100 DO
            OUTPUT ErrMessage
            INPUT n
    END_WHILE
    sum = 0
    FOR i = 1 TO n-1 DO
            IF n MOD i = 0 THEN
                    sum = sum + i
            END
    END_FOR
    IF sum = n THEN
            DISPLAY n + " is perfect number"
    ELSE
            DISPLAY n + " not is perfect number"
    END_IF
END
```

- C code

```
/*Program to input positive integer n and less than or equal
100.
Display to the screen n is perfect number or not
date writen:29.06.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>

void main(void)
{
	//declare variable
	int n;
	int i;
	int sum;
	//Clear screen
	clrscr();
	printf("\nEnter a positive integer please:");
	scanf("%d",&n);
	while(n<=0 || n>100)
	{
		printf("\nPlease reenter n so that n greater 0 and
less than or equal 100:");
		scanf("%d",&n);
	}
	//Calculate sum of all divisors except it.
	sum=0;
	for(i=1;i<n;i++)
		if(n%i==0)
		{
```

```
                sum=sum+i;
           }

      if(sum==n)
           printf("\n%d is a perfect number",n);
      else
           printf("\n%d is not a perfect number",n);

      printf("\nPress any key to continue");
      getch();//stop screen to view result
}
```

Exercise 4: Write a C program to display to the screen stars likes figure below.

```
*
* *
* * *
* * * *
* * * * *
```

- Solution: According to the figure we find out that row one has one star, row two have two stars, row three have three stars, and so on. Finally, we find out the rule: the number of star in the row equals to this row.

- Pseudo code:

```
  BEGIN
    FOR row = 1 TO 5 DO
        FOR col = 1 to row DO
            DISPLAY “*”
        END_FOR

        MOVE cursor to new line

    END_FOR

  END
```

```
/*Program to display to the screen rectangle of stars.
date writen:29.06.2008
author:
version:1.0*/
#include<stdio.h>
#include<conio.h>

void main(void)
{
     //declare variable
     int row;
     int col;
     //Clear screen
     clrscr();
     for(row=1;row<=5;row++)
     {
          for(col=1;col<=row;col++)

               printf("*");
          printf("\n");
     }
     printf("\nPress any key to continue");
     getch();//stop screen to view result
}
```

II. EXERCISES WITHOUT SOLUTION

Exercise 1: Write a C program inputs integer n so that n greater than or equal 10 and less than or equal 20. Display to the screen the rectangle of stars base on given n.

Example: if n was inputted 10 then the triangle of stars display on the screen like below:

```
*
```

```
* *

* * *

* * * *

* * * * *

* * * * * *

* * * * * * *

* * * * * * * *

* * * * * * * * *

* * * * * * * * * *
```

Exercise 2: Write a C program inputs positive integer n, calculate and display total all prime numbers from 1 to n.

Hint: Scanning from 1 to n to find out all prime numbers and calculate accumulate total.

Exercise 3: Write a C program inputs positive integer n, display to the screen all perfect number from 1 to n.

Hint:

- Input and validate n.
- Scanning from 1 to n to find and display all perfect numbers

Exercise 4: Write a C program produces menu like that:

1. Enter two integers
2. Calculate sum of two given integers
3. Calculate subtraction of two given integers
4. Calculate multiply of two given integers

5. Calculate division of two given integers
6. Exit

Enter your choice:

When user chooses 1: Input two integers from keyboard; chooses 2: Calculate and display total of two given integers; chooses 3: Calculate and display subtraction of two given integers; chooses 4: Calculate and display multiply of two given integers; chooses 5: Calculate and display division of two given integers; chooses 6: Exit from program.

Hint: Using outline Pseudo code below to solve menu problem

```
REPEAT
    DISPLAY Menu
    INPUT Choice
    CASE Choice OF
        CASE 1: INPUT two integers
        CASE 2: Calculate and display addition of two given numbers
        CASE 3: Calculate and display subtraction of two given numbers
        CASE 4: Calculate and display multiply of two given numbers
        CASE 5: Calculate and display division of two given numbers
    END_CASE
UNTIL Choice = 6
```

Exercise 5: A program is required to calculate the cost of parking in a car park. The program is to accept the input of the number of hours parked in the car park (rounded up) and display the cost based on the following rates:

1 to 4 hours RATE1=$3 per hr

5 to 7 hours RATE2=$1.50 per hr>=5 hrs + 4 hrs*RATE1

8 to 24 hours FLATRATE=$18

The program is to repeatedly prompt for the input of hours until a value of 0 is entered. When a value of 0 is entered for the number of hours, the program should display the following statistics:

- The total number of cars for each time period.
- The total number of hours for all cars processed.
- The total of all car-parking fees collected.
- The average number of hours a car stayed in the car park.
- The average cost of a stay in the car park.

The program should operate correctly for a bad day when no cars use the car park

Hint: Using outline pseudo code below to solve sentinel problem

```
ASSIGN statistic variable to zero
INPUT Hour
WHILE Hour <> 0 DO
    Calculate rate
    Calculate fee
    Update statistics and accumulate variable
    INPUT Hour
END_WHILE
DISPLAY statistics and accumulate variable
```

Exercise 6: Write a C program is to read a collection of integer data items and find and print the index of the first occurrence and the last occurrence of the number 24. The program will print index value of 0 if the number 24 is not found. The index is the sequence number of the data item 24. For example, if the fifth data item is the only 24, then the index value 5 should be printed for the first and last occurrence. The program should prompt the user whether they want to continue processing or not. If the answer is 'N' then the result will be displayed. If 'Y' then continue processing. Relevant data validation and error messages should be considered in the program.

Hint: Using outline pseudo code below to solve conversation problem

```
Assign first and last index, counter to zero
REPEAT
    INPUT Integer
    Increase counter
    Check integer if it equals to 24 then
    Check counter if it equals to 1 then
        Assign first and last index equals to counter
        Otherwise only assign last index equals to counter
    Input and validate Answer so that it must be ‘Y’ or ‘N’
UNTIL Answer = ‘N’
DISPLAY first and last index
```