Exercise Problems - 5

Break and Continue

1. Write a program (without using libraries other than `stdio.h`) to take a number n as input from the user and print the list of all numbers which are perfect squares less than n. That is, if $n = 25$, the output should be the list $1, 4, 9, 16$. Can you do this program so that multiplication is used only in one instruction in your program? (Clue: use an infinite loop and a break instruction.)

2. Write a program that repeatedly asks the user to enter numbers until she/he gives 0 or a negative number as input. Once the user gives 0 or a negative number, the maximum of all numbers entered till then should be printed as output to the user. (Note that the user may never stop entering positive values. In that case, your program should not stop. If the first value entered is zero or negative, you can give zero as output to the user.)

3. Write a C program that takes a number n, followed by a set of n numbers from user and does the following: a) stores the list of numbers in an array b) search the array to see if any multiples of 5 are there in the list and if yes, print the position of the first occurance of a multiple of 5 in the list. If no multiples of 5 are found, tell the user that there are no multiples of 5 in the list.

4. Rewrite the above program so that it prints the position of the last occurance of a multiple of 5 in the list instead of the first occurance. Reduce the number of comparisions performed by your program, by searching from the end of the list.

5. Read the following program and try to understand what it does. Execute the program giving inputs n=20 and 1,2,...,20 as the list of elements. Is your understanding of the program correct?

```
#include<stdio.h>
int main()
{
    int a[20], n, counter, n8, n4, n2;
    printf("enter n (<=20) \n");
    scanf("%d",&n);
    if(n>20)
    {
       printf("wrong input \n");
    }
    else
    {
        counter = 0;
        while (counter < n)
        {
```

```
            printf("enter the next element \n");
            scanf("%d",&a[counter]);
            counter = counter +1;
          }
          counter = -1;
          n8=0; n4=0; n2=0;
          while (counter < n-1)
          {
             counter=counter+1;
             if (a[counter]%8==0)
             {
                 n8=n8+1;
                 continue;
             }
             if (a[counter]%4==0)
             {
                 n4=n4+1;
                 continue;
             }
             if (a[counter]%2==0)
             {
                 n2=n2+1;
             }
          }
          printf("n8=%d, n4=%d, n2=%d \n", n8, n4,n2);
      }
      return(0);
}
```