KATHMANDU UNIVERSITY

Department of Computer Engineering

A

Lab Report On

Computer Programming {COMP 102}

Lab Sheet No: 02

Submitted by:
Ashraya Kadel
UNG CE I/I
Roll No: 25

Submitted to:
Sameer Tamrakar
Department of Computer Engineering

Submission date: 05/03/2023

WEEK 5: LOOPING

In week 5, we learnt about the use of loops and looping statement while writing programs.

<Q.1> Write a program to read a sentence and counts the total numbers of characters (excluding space) using while loop.

Ans:

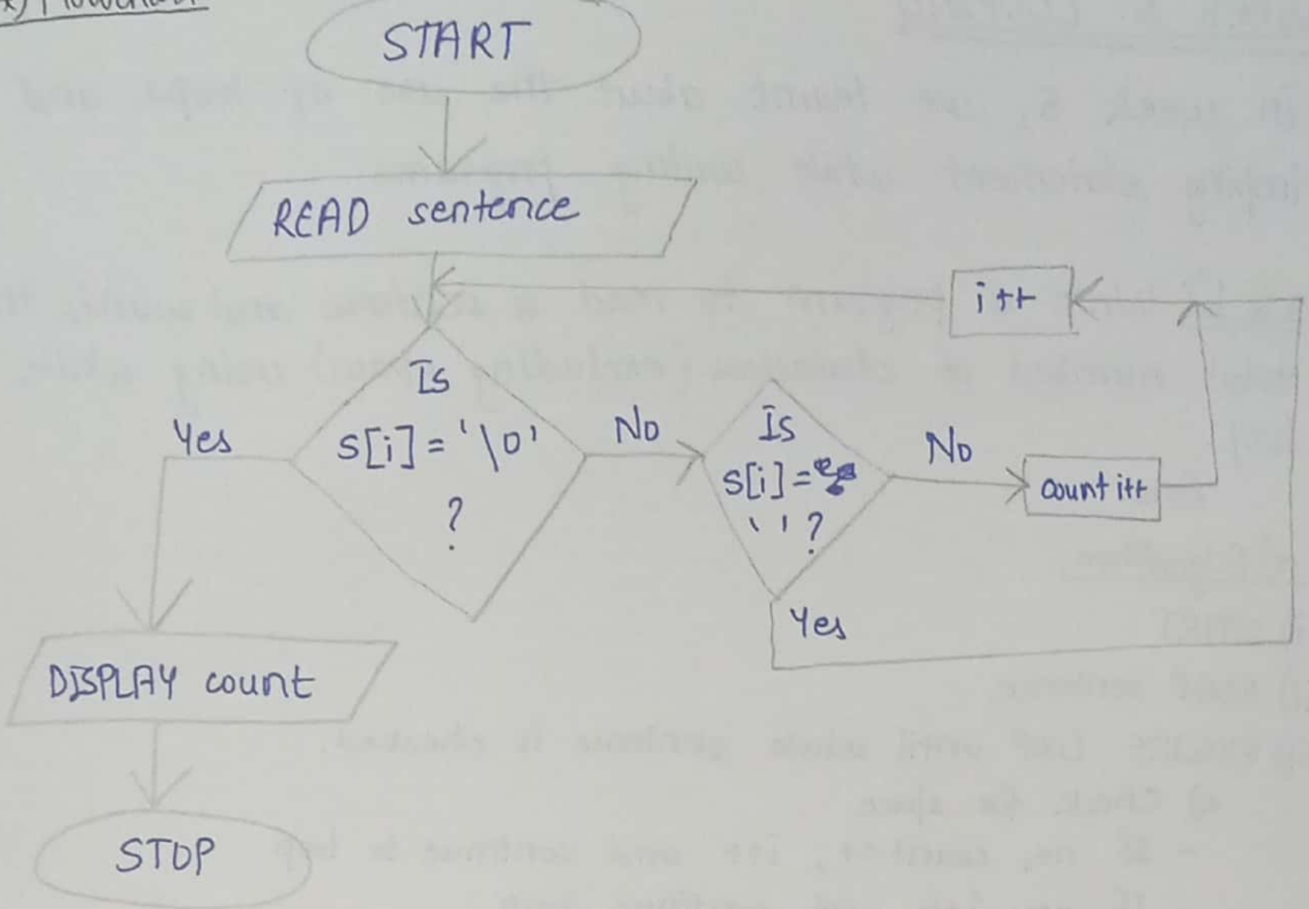
*) Algorithm

```
i) START
ii) READ sentence
iii) EXECUTE LOOP until whole sentence is checked.
    a) Check for space
        - If no, count ++, i++ and continue to loop
        - If yes, i++ and continue loop
    b) End loop when the character is returned as null.
iv) DESPLAY number of characters
v) STOP
```

*) Source Code

```
# include<stdio.h>
# include <string.h>
void main()
{
char s[70];
printf("Enter a sentence 70 characters long");
gets (s);
int i=0, count=0;
while (s[i]!='\0')
{
if (s[i]!=' ')
{count++;}
```

*) Flowchart


START
READ sentence
i++
Is
S[i] = '\0'
?
Yes
No
Is
S[i] = ' ' ?
No
count i++
Yes
DISPLAY count
STOP

```
i++;
}
printf("No.of characters in sentence = %d\n", count);
}
```

*) Output

```
Enter a sentence 70 characters long
My name . is Kadel
No.of characters in sentence = 13
```

*) Description:

This program reads a sentence and checks for the number of characters in the sentence without space. It displays the number of characters. using while loop.

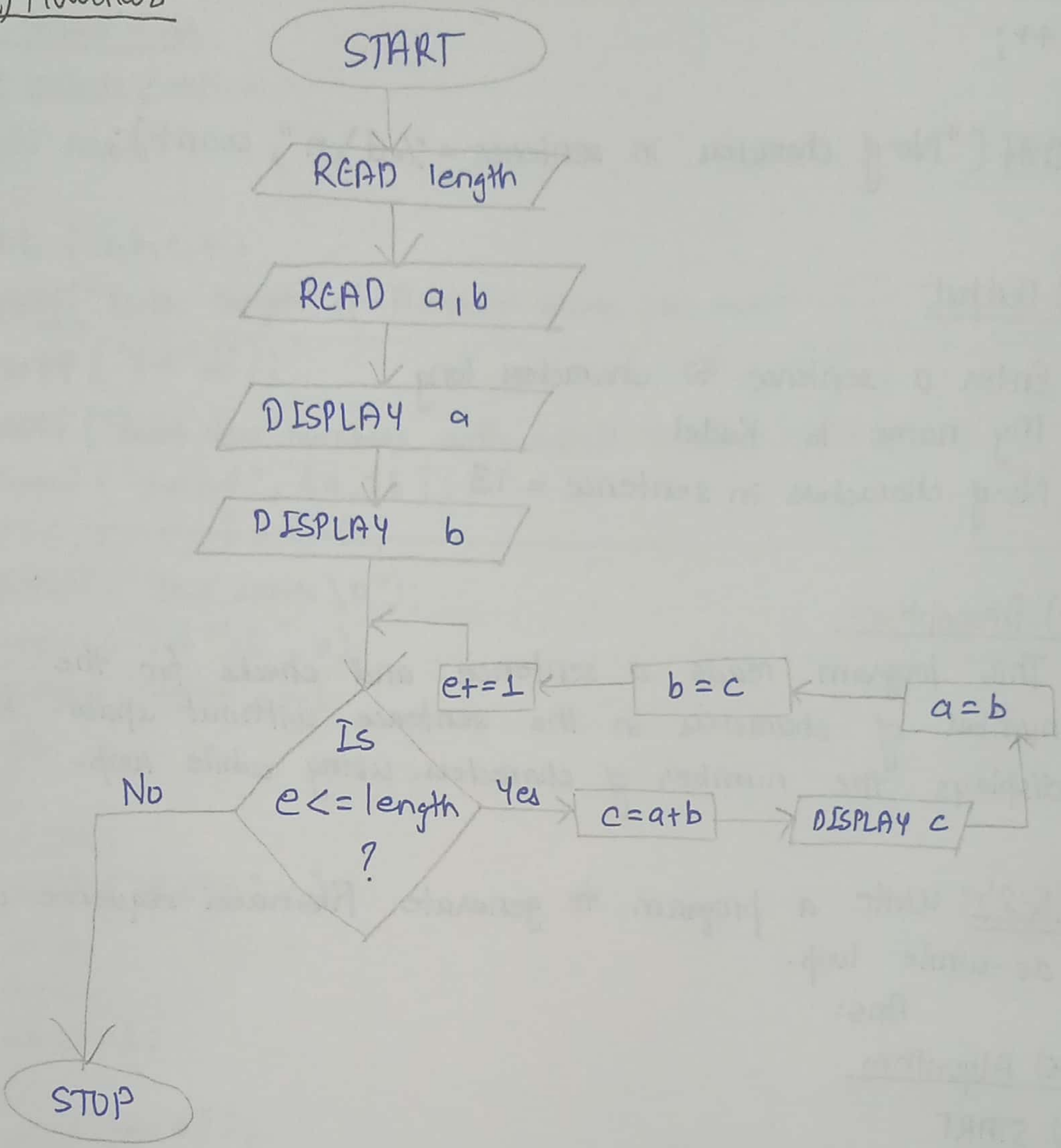
<Q.27>: Write a program to generate fibonacci sequence using do-while loop.

Ans:

*) Algorithm

```
i) START
ii) READ length of fibonacci series (i)
iii) READ two numbers to start series with (a,b).
iv) DISPLAY two numbers
v) EXECUTE DO LOOP
        a) c = a + b
        b) DISPLAY c
        c) Change a to b, b to c
        d) END loop if inputted length of series is achieved.
vi) STOP
```

*) Flowchart


START
READ length
READ a,b
DISPLAY a
DISPLAY b
Is
e<=length
?
No
Yes
c=a+b
DISPLAY c
a=b
b=c
e+=1
STOP

*) Source Code

```
# include <stdio.h>
void main()
{
int i,a,b,c,e;
printf("Enter length of Fibonacci series you want\n");
scanf ("%d",&i);
printf ("Enter two numbers with which to start series\n");
scanf ("%d%d", &a,&b);
c=0; e=0; i=i-2;
printf("Your series\n");
printf("%d\t", a);
printf ("%d\t", b);
do
{
c=a+b;
printf ("%d\t",c);
a=b;
b=c;
e+=1;
}
while (e<i);
}
```

*) Output

```
Enter length of fibonacci series
6
Enter two numbers with which to start series
1
1
Your series
1   1   2   3   5   8
```

*) Description

This program reads the length of Fibonacci series and the two numbers to start series with. The loop is executed for as to obtain the inputted length of Fibonacci series.

<Q-3> Write a program to read number and identify whether given number is a prime number or not.

Ans:

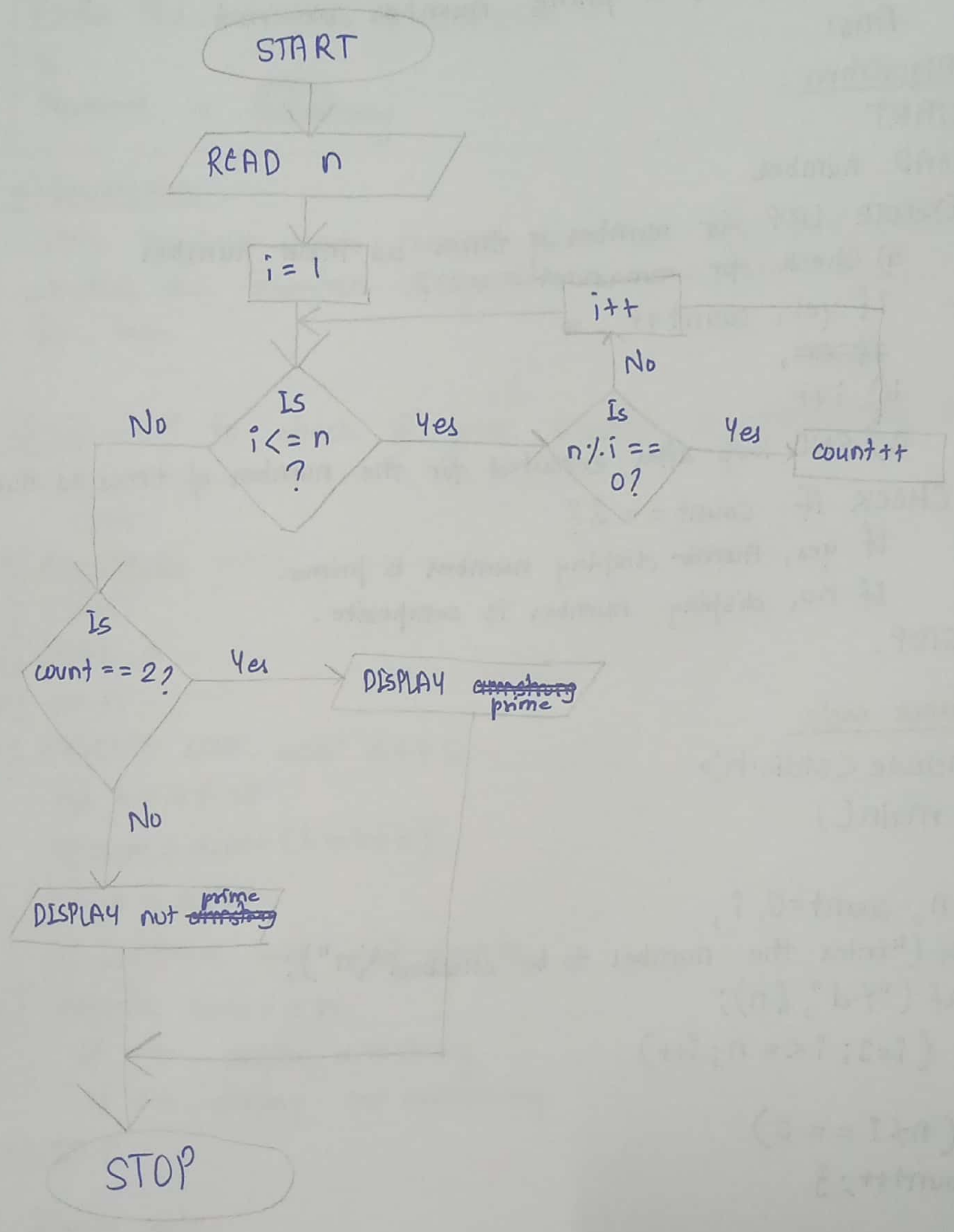
*) Algorithm :

```
i) START
ii) READ number
(iii) EXECUTE LOOP for number of times as input number
        a) Check for remainder == 0?
           If yes, count++ , ~~co~~
           ~~if no~~,
        b) i++
        c) END loop after executed for the number of times as numbers.
(iv) CHECK if count == 2?
        If yes, ~~Numb~~ display number is prime.
        If no, display number is composite.
(v) STOP.
```

*) Source code

```
#include <stdio.h>
void main()
{
int n, count=0, i;
printf("Enter the number to be checked \n");
scanf("%d", &n);
for (i=1; i<= n; i++)
{
if (n%i == 0)
{ count++; }
}
if (count== 2)
{ printf(" Number is ~~armstrong~~ prime"); }
else
{ printf("Number is ~~not armstrong~~ composite"); }
}
```

(*) Flowchart


START
READ n
i = 1
i++
No
Is i<=n ?
No
Yes
Is n%i == 0?
Yes
count++
Is count == 2?
Yes
DISPLAY ~~armstrong~~ prime
No
DISPLAY not ~~armstrong~~ prime
STOP

*) Output

```
Enter the number to be checked
5
Number is prime ~~armstrong~~
```

*) Description :

This program reads numbers from the user and checks whether the number is prime or composite using for loop.

<Q.47>: WAP to check if given number is armstrong or not.

Ans:

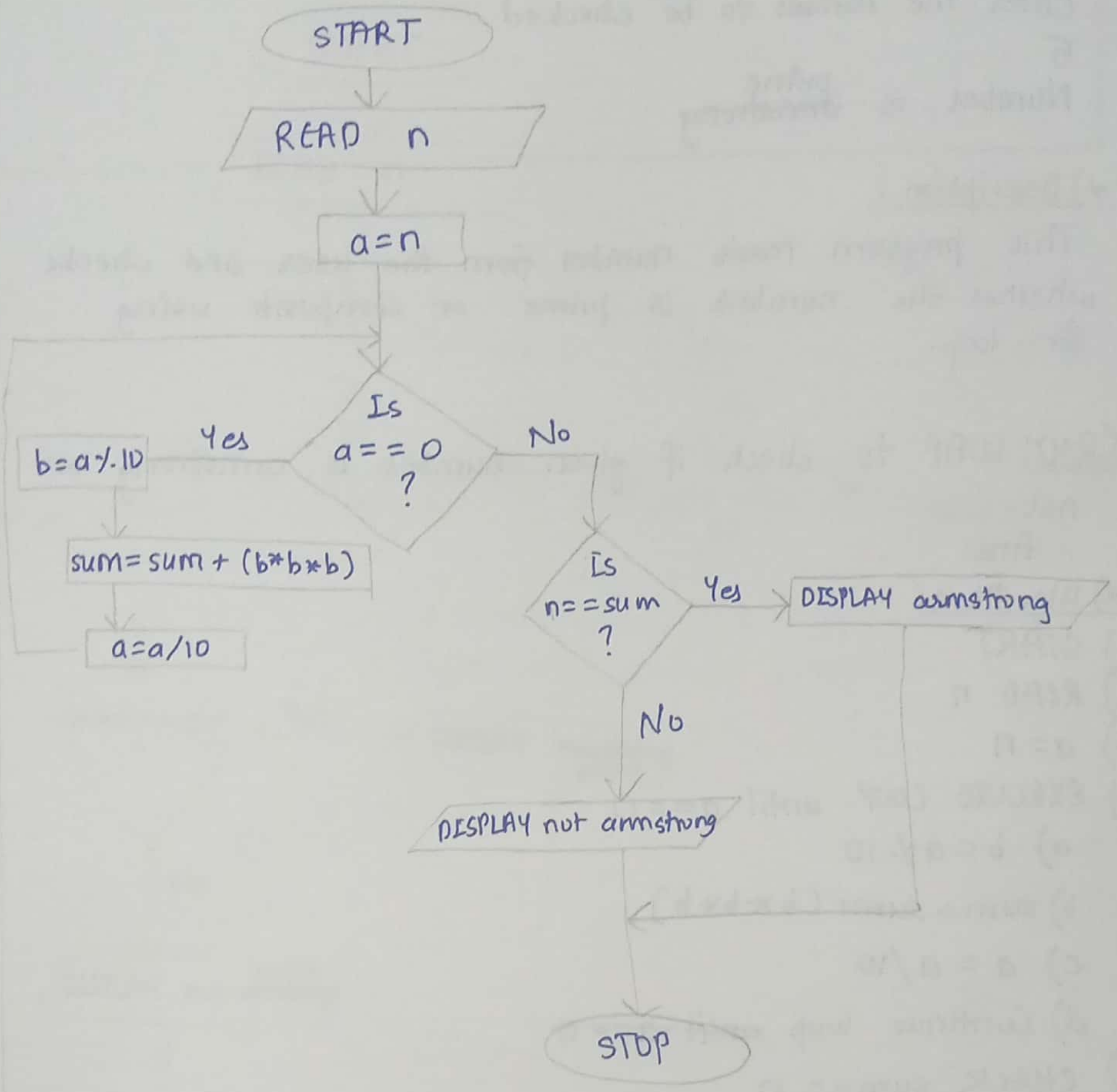
*) Algorithm:

```
i) START
ii) READ n
iii) a = n
iv) EXECUTE LOOP until a == 0
    a) b = a % 10
    b) sum = sum + (b * b * b)
    c) a = a / 10
    d) Continue loop ~~until a == 0~~
v) CHECK sum == n
    If yes, display armstrong
    If no, display not armstrong
vi) STOP
```

*) Source code:

```
# include <stdio.h>
void main()
{ int a, n, sum = 0, b = 0;
printf("Enter a number\n");
scanf("%d", &n);
a = n;
```

*) Flowchart


START
READ n
a=n
Is a==0 ?
Yes
b=a%10
sum=sum+(b*b*b)
a=a/10
No
Is n==sum ?
Yes
DISPLAY armstrong
No
DISPLAY not armstrong
STOP

```
~~for ( while (a!=0~~
while (a!=0)
{
 b= a% 10;
 sum = sum + (b*b*b);
  a = a/10
 }
 if (sum == n)
 {printf(" Number is armstrong") ;}
 else
{printf("Number is not armstrong"); }
 }
```

*) Output

```
Enter a number
371
It is armstrong
```

*) Description

This program reads a number and checks whether it is armstrong or not. We use while loop to check for armstrong. Here, the number entered as 371 is checked and confirmed as being armstrong.

<Q.5> WAP to check whether the given number is perfect number or not?

Ans:

*) Source code:

```
#include <stdio.h>
void main ( )
{
int a, b, sum=0, i;
a =0;
printf (" Enter a number to check \n");
scanf ("%d", &b);
 a= b/2 ;
for (i=1; i<= a; i++)
 {
 if (b% i == 0)
 { sum = sum + i;}
 }
if (sum = n)
{printf (" Number is perfect");}
else
{printf ("Number is not perfect");}
}
```

*) Output

```
Enter a number to check
6
Number is perfect
```

<Q.6>: WAP to check factorial of given number.
Ans:

*) Source code

```
#include <stdio.h>
void main()
{
int n, fact=1, i;
printf("Enter the number\n");
scanf("%d", &n);
for (i=1; i<=n; i++)
{
 fact = fact * i
 }
printf("The factorial = %d\n", fact);
}
```

*) Output

```
Enter the number
5
The factorial = 120
```

<Q.7> WAP to check if given number is dudney or not.

Ans:

*) Source code:

```
# include <stdio.h>
# include <math.h>
void main ()
{
 int n, a=0, sum=0, b=0, c=0
 printf ("Enter a number");
 Scanf ("%d", &n);
 a=n;
 while (a!=0)
 {
   b = a%10;
   sum = sum+b;
   a=a/10
 }
 C= pow (sum, 3);
 if (c==n)
 { printf ("Number is dudeney"); }
 else
 { printf ("Number is not dudney"); }
}
```

*) Output

```
Enter a number
4913
Number is dudeney
```

⟨Q.8⟩ Generate the following patterns

i): 54321
    4321
    321
    21
    1

Ans:

*) Source code:

```
#include <stdio.h>
void main ( )
{
int i,j;
for (i=0; i<=4; i++)
{
for (j=5; j>i; j--)
{
printf ("%d", j-i);
~~printf ("% t)~~ printf ("\t");
}
printf ("\n");
}
}
```

*) Output

```
5   4   3   2   1
4   3   2   1
3   2   1
2   1
1
```

(ii)

```
1
1 2
1 2 3
1 2 3 4
```

Ans:

*) Source code:

```
# include <stdio.h>
void main ( )
{
int i,j ;
for (i=1; i<=4; i++)
{
for (j=1 ; j <= i, j++)
{
printf("%d",j);
printf ("\t");
}
printf ("\n");
}
}
```

*) Output

```
1
1    2
1    2    3
1    2    3    4
```

(iii)

```
      1
    1 2
  1 2 3
1 2 3 4
```

Ans:

*) Source code:

```
# include <stdio.h>
void main ()
{
int i, j, k;
for (i=1; i<=4; i++)
{
for (k=0 ; k<4-i ; k++)
{
printf(" ");
}
for (j=1; j<=i ; j++)
{
printf ("%d", j);
printf ("\t");
}
printf ("\n");
}
}
```

*) Output

```
      1
    1 2
  1 2 3
1 2 3 4
```

(iv):

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

Ans

*) Source code :

```
#include <stdio.h>
void main ()
{
int i,j,k;
for (i=1; i<=4; i++)
{
for (k=0; k<4-i; k++)
{
printf (" ");
}
for (j=1; j<=((2*i)-1); j++)
{
printf ("*");
printf ("\t"); }
printf ("\n");
}
}
```

*) Output

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

```
(v): 5 4 3 2 1
     4 3
     3 2 1
     2 1
     1
```

Ans:

*) Source code:

```
#include <stdio.h>
void main()
{
int i, j;
for (i=0; i<=4; i++)
{
for (j=5; j>i; j--)
{
if (i==1 && j==3)
{break; }
printf("%d", j-i);
printf ("\t");
}
printf("\n");
}
}
```

*) Output

```
5   4   3   2   1
4   3
3   2   1
2   1
1
```

(vi):

```
5 4 3 2 1
4 3 1
3 2 1
2 1
1
```

Ans:

*) Source code:

```
#include <stdio.h>
void main ()
{
int i,j;
for (i=0; i<=4; i++)
{
for (j= 5; j>i, j--)
{
if ( i==1 && j==3 )
{continue; }
printf ("%d", j-i);
printf (" \t ");
}
printf ("\n");
}
}
```

*) Output:

```
5 4 3 2 1
4 3   1
3 2 1
2 1
1
```