

Q1) Write an algorithm and C program to find the factorial of number.

Algorithm

Step 1: Start
Step 2: Read N
Step 3: Initialize i=1
Step 4: Initialize f=1
Step 5: Fact = fact * i
Step 6: i = i + 1
Step 7: Check $i \leq n$
 if true goto step 5
 if false goto step 8
Step 8: Display fact
Step 9: Stop

C Program

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

2 Write an algorithm and program in C to find the given no. is odd or even

Algorithm

Step 1 : Start

Step 2 : Enter a number

Step 3 : If number mod 2 = 0 then

 Print even number

else goto step 4

Step 4 : Print odd number

Step 5 : Stop

C Program

```
#include<stdio.h>
int main()
{
    int n, odd, even;
    printf("Enter a number:");
    scanf("%d", &n);
    if (n%2 == 0)
    {
        printf("The number is even: %d", even);
    }
    else
    {
        printf("The number is odd: %d", odd);
    }
}
```

```
* C:\Users\Apple\Documents\fact.exe
enter a number:8
the number is even:0
Process exited after 4.841 seconds with return value 0
Press any key to continue . . .
```

3) Write an algorithm and a C Program to print sum of natural numbers.

Algorithm

Step 1 : Start
Step 2 : Read N
Step 3 : Sum = $(n + (n+1))/2$
Step 4 : Display sum
Step 5 : Stop

C Program

```
#include <stdio.h>
int main()
{
    int n, sum = 0;
    printf("Enter number:");
    scanf("%d", &n);
    sum = (n * (n + 1)) / 2;
    printf("The sum is: %d", sum);
}
```

```
[1] C:\Users\Apple\Documents\fact.exe
enter a number:8
the sum is:36
Process exited after 1.627 seconds with return value 0
Press any key to continue . . .
```

Q) Write a program in C and an algorithm to check given no. is positive or negative.

Algorithm

Step 1: Start

Step 2: Take number n

Step 3: If ($n > 0$) then

 display "Positive Number"

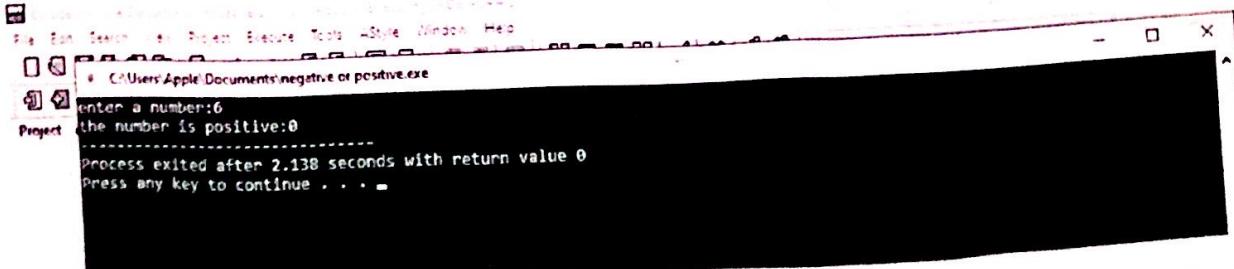
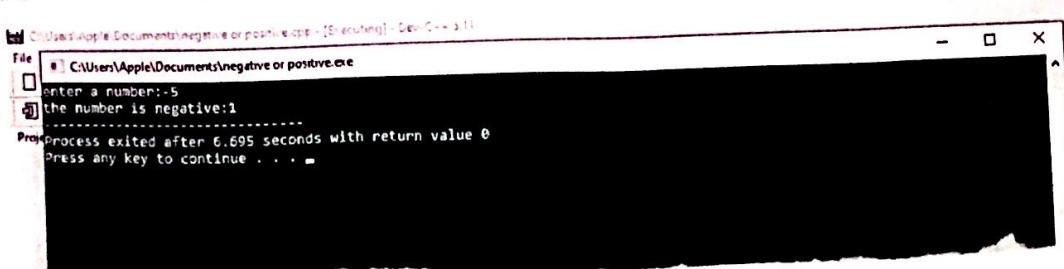
Otherwise goto step 4

Step 4: Display "Negative Number"

Step 5: Stop

C Program

```
#include <stdio.h>
int main()
{
    int n, positive, negative;
    printf("Enter a number:");
    scanf("%d", &n);
    if(n > 0)
    {
        printf("The number is positive : %d", positive);
    }
    else
    {
        printf("The number is negative : %d", negative);
    }
}
```



5) Write an algorithm and a C Program whether the number is prime or not.

Algorithm

Step 1: Start
Step 2: Read 'n' to check prime or not
Step 3: Set $i=1$, count = 0
Step 4: If $i \leq n$,
 if true goto step 5
 else goto step 8
Step 5: If $n \% i == 0$
 if true goto step 6
 else goto step 7
Step 6: Set count = count + 1
Step 7: $i = i + 1$, go to step 4
Step 8: Check count, if count = 2,
 display 'Prime'
 Else it is 'not Prime'
Step 9: Stop

C Program

```
#include <stdio.h>
int main()
{
    int i, count = 0, n, prime;
    printf("Enter a number:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
        if(n % i == 0)
    {
        count = count + 1;
    }
    if(count == 2)
    {
        printf("The number is prime: %d", prime);
    }
    else
    {
        printf("The number is not prime: %d", not prime);
    }
}
```

6) Write an algorithm and a C Program to display Name, Address and Phone number.

Algorithm

- Step 1: Start
- Step 2: Declare variables name, address and Phone number
- Step 3: Input values of name, address and Phone number
- Step 4: Print name, address and Phone number
- Step 5: Stop

C Program

```
#include <stdio.h>
int main()
{
    int phonenumber;
    char name, address;
    printf("Enter phone number:");
    scanf("%d", &phonenumber);
    printf("Enter name and address:");
    scanf("%s%s", &name, &address);
    printf("name is: %s\n", name);
    printf("address is: %s\n", address);
    printf("phonenumber is: %d\n", phonenumber);
}
```

```
* C:\Users\Apple\Documents\prime number.c
enter phonenumber:9887543210
enter name and address:Alisha New Baneshwar
-----
Process exited after 20.62 seconds with return value 3221225477
Press any key to continue . . .
```

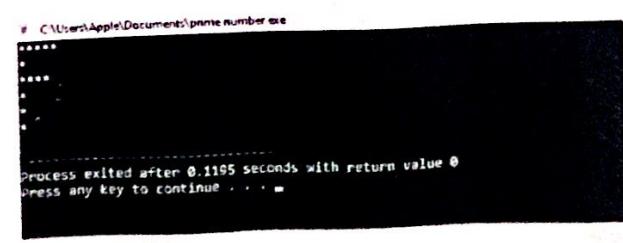
7) Write an algorithm and C Program to display F Pattern.

Algorithm

Step 1: Start
Step 2: Read * * * * * |n
Step 3: Read * |n
Step 4: Read * * * * |n
Step 5: Read * |n
Step 6: Read * |n
Step 7: Read * |n
Step 8: Display * * * * * |n
Step 9: Display * |n
Step 10: Display * * * * |n
Step 11: Display * |n
Step 12: Display * |n
Step 13: Display * |n
Step 14: Stop

C Program

```
#include <stdio.h>
int main()
{
    printf("* * * * * |n");
    printf(" * |n");
    printf(" * * * * |n");
    printf(" * |n");
    printf(" * |n");
    printf(" * |n");
}
```



8) WAP In C and an algorithm to find area and perimeter of a circle.

Algorithm

Step 1: start

Step 2: Read radius 'r' of a circle

Step 3: Area = $(\pi \times r^2)$

Step 4: Perimeter = $(2 \times \pi \times r)$

Step 5: Display Area of a circle

Step 6: Display perimeter of a circle

Step 7: stop

C Program

```
# include <stdio.h>
int main()
{
    float r, area, perimeter;
    printf("enter radius:");
    scanf("%f", &r);
    area = (4 * 3.14 * (r * r));
    perimeter = (2 * 3.14 * r);
    printf("The area is :%.f", area);
    printf("The perimeter is :%.f", perimeter);
}
```

```
C:\Users\Apoli\Documents\univ number\cse
enter radius:5
the area is:78.540000the perimeter is:31.420000
.....
process exited after 0.145 seconds with return value 0
press any key to continue . . .
```

g) Write an algorithm and a C Program to find area and perimeter of rectangle.

Algorithm

- Step 1 : Start
- Step 2 : Read length 'l' and breadth 'b' of rectangle
- Step 3 : Area = $l \times b$
- Step 4 : Perimeter = $2(l + b)$
- Step 5 : Display the area of rectangle
- Step 6 : Display the perimeter of rectangle
- Step 7 : Stop

C Program

```
#include<stdio.h>
int main()
{
    int l, b, area, perimeter;
    printf("Enter length and breadth:");
    scanf("%d%d", &l, &b);
    area = l * b;
    perimeter = 2 * (l + b);
    printf("The area is: %d", area);
    printf("The perimeter is: %d", perimeter);
}
```

```
* C:\Users\Apple\Documents\prime numbers.exe
Enter length and breadth:7 9
the area is:63the perimeter is:32
-----
Process exited after 5.447 seconds with return value 0
Press any key to continue . . .
```

10) Write an algorithm and program in C to convert given number into year month and day.

Algorithm

- Step 1 : Start
- Step 2 : Read the number of days
- Step 3 : $\text{years} = \text{Years} / 365$
- Step 4 : $\text{days} = (\text{days} \% 365) / 7$
- Step 5 : ~~$\text{days} = \text{days} - (365 * \text{years}) + (7 * \text{weeks}) * (\text{days} \% 365) \% 7$~~
- Step 6 : Display no. of years
- Step 7 : Display no. of weeks
- Step 8 : Display no. of days
- Step 9 : Stop

C Program

```
#include <stdio.h>
int main()
{
    int n, days, weeks, years;
    printf("Enter number of days:");
    scanf("%d", &n);
    years = n / 365;
    weeks = (n % 365) / 7;
    days = (n % 365) % 7;
}
printf("years is %d : %.d\n", years);
printf("weeks is %d : %.d\n", weeks);
printf("days is %d : %.d\n", days);
return 0;
```

Q) Write an algorithm and a C Program to swap the given number.

Algorithm

Step 1: Start
Step 2: Read number of
Step 3: $a = a + b;$
Step 4: $b = a - b;$
Step 5: $a = a - b;$
Step 6: Display number b and a
Step 7: Stop

C Program

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter numbers:");
    scanf("%d%d", &a, &b);
    printf("before swapping a: %d\n and b : %d\n", a, b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("after swapping a: %d\n and b: %d\n", a, b);
    return 0;
}
```

```
* C:\Users\Apple\Documents\Number of days.exe
enter numbers:12 13
before swapping a:12
and b:13
after swapping a:13
and b:12
Process exited after 4.248 seconds with return value 0
Press any key to continue . . .
```

Write an algorithm and a C Program to find sum of no. of digits.

Algorithm

Step 1: Start
Step 2: Read number 'n'.
Step 3: $r_1 = n \% 10;$
Step 4: $n = n / 10;$
Step 5: $r_2 = n \% 10;$
Step 6: $n = n / 10;$
Step 7: $r_3 = n \% 10;$
Step 8: $sum = r_1 + r_2 + r_3$
Step 9: Display sum
Step 10: Stop

C Program

```
#include <stdio.h>
int main()
{
    int n, r1, r2, r3, sum;
    printf("Enter a number:");
    scanf("%d", &n);
    r1 = n % 10;
    n = n / 10;
    r2 = n % 10;
    n = n / 10;
    r3 = n % 10;
    sum = r1 + r2 + r3;
    printf("The sum is: %d\n", sum);
}
```

Q) Write an algorithm and a C Program to find sum of no. of digits.

Algorithm

Step 1: Start
Step 2: Read number 'n'.
Step 3: $r_1 = n \% 10;$
Step 4: $n = n / 10;$
Step 5: $r_2 = n \% 10;$
Step 6: $n = n / 10;$
Step 7: $r_3 = n \% 10;$
Step 8: $sum = r_1 + r_2 + r_3$
Step 9: Display sum
Step 10: Stop

C Program

```
#include <stdio.h>
int main()
{
    int n, r1, r2, r3, sum;
    printf("Enter a number:");
    scanf("%d", &n);
    r1 = n % 10;
    n = n / 10;
    r2 = n % 10;
    n = n / 10;
    r3 = n % 10;
    sum = r1 + r2 + r3;
    printf("The sum is: %d\n", sum);
}
```

```
1 C:\Users\Apple\Documents\sum of digits.exe
2 enter a number:765
3 sum is:18
4 -----
5 Process exited after 2.749 seconds with return value 0
6 any key to continue
```

1.5.2.3 Repetition

Repetition can be implemented using loops.

The repeat loop is used to iterate some condition becomes true. Its general form is:

Repeat
Process1
Process2
End

- The ...

- The logical operators (AND, OR, NOT)

3 Flowcharting

Flowchart is the graphical representation of an algorithm.

In other words, flowchart is a pictorial representation of an algorithm.

(3) Write an algorithm and a C Program to find factors of a given number.

Algorithm

Step 1: Start
Step 2: Read number 'n'.
Step 3: Initialize ($i=1$).
Step 4: $i = i + 1$. If ($n \% i = 0$)
Step 5: if true goto step 6
if false go to step 8
Step 6: $i = i + 1$
Step 7: Check $i \leq n$
if true goto step 4
if false go to step 8
Step 8: Display factors
Step 9: Stop

C Program

```
#include <csstdio.h>
int main()
{
    int i, n;
    printf("Enter a number:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        if(n%i==0)
        {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

Q5) Write an algorithm to print even numbers between 1 - 100 and a C Program

Algorithm

Step 1: Start

Step 2: Set $i = 1$

Step 3: Repeat step 4 to 5 until $i <= 100$ do

Step 4: if $i \% 2 == 0$ then

 print: i

Step 5: Set $i = i + 1$

Step 6: Stop

C Program

```
#include <stdio.h>
int main()
{
    int i, even;
    for(i=1; i<=100; i++)
        if(i%2==0)
            printf("The even numbers are : %d", i, even);
}
```

Algorithm

Write an algorithm and a C Program to print odd numbers between 1-100.

Step 1: Start
Step 2: Set $i = 1$
Step 3: Repeat step 4 to 5 until $i \leq 100$ do
Step 4: if $i \% 2 \neq 0$ then
 print : i
Step 5: Set $i = i + 1$
Step 6: Stop

Program

```
include <stdio.h>
int main()
{
    int i, odd;
    for(i=1; i<=100; i++)
        if(i%2 != 0)
            printf("The odd numbers are: %d", i, odd);
}
```