

Jadavpur University

*Computer Science
and
Engineering Department*

Name	Tathagata Sur
Roll No.	002310501030
Class	BCSE
Year	1 st Year

*Computer Programming
and
Numerical Methods*

INDEX

Sl. No.	Topic	Program	Page No.	Signature
1.	I/O Conditional	1. 2. 3. 4. 5. 6. 7. 8.	1 1, 2 2, 3 3 3, 4 4, 5 5, 6 6	
2.	Flowchart and Algorithm	1. 2. 3. 4.	7 8 9 10	
3.	Loops	1. 2. 3. 4. 5. 6.	11, 12, 13 14, 15 16 16 17 18, 19, 20	

4.	Array	1. 2. 3. 4. 5. 6.	21 21, 22 23 23, 24 24, 25 25, 26	
5.	Function And Pointer	1a. 1b. 2. 3. 4. 5. 6.	26, 27 27 27, 28 28 29, 30 30 30, 31	
6.	Files	1. 2.	31, 32 33	
7.	Numerical Methods	1. 2.	34 35	

I/O and Conditional Statements

Q1. Write a C program that reads two values from the keyboard, swaps their values and prints out the result.

```
#include <stdio.h>

int main()
{ int a, b, temp;
  printf("Enter 1st no.:");
  scanf ("%d",&a);
  printf("Enter 2nd no.:");
  scanf ("%d",&b);
  temp=a ; a=b ; b=temp;
  printf(" 1st no.=%d", a);
  printf ("2nd no.=: %d", b) ;
  return 0;
}
```

Input phase

Enter 1st no. : 2

Enter 2nd no. : 3

Output phase

1st no. = 3

2nd no. = 2

Q2. If a three-digit integer is input through the keyboard, write a C program to calculate the sum of its digits (Hint: Use the modulo operator ")

```
#include <stdio.h>

int main()
{
    int c, sum=0;

    printf("Enter a 3 digit no. :");

    scanf("%d",&c);

    while (c>0)
    {
        sum+=(c%10);
        c/=10;
    }

    printf(" Sum of Digits=%d", sum);

    return 0;
}
```

Input

Enter a 3 digit no. : 111

Output

Sum of digits = 3

Q3. Write a C program which accepts basic salary as input and prints the gross salary, which is the sum of the basic, dearness allowance (60% of basic salary), and house rent allowance (15% of basic salary).

```
#include <stdio.h>

void main()
{
    int n;

    printf("Enter basic salary :");

    scanf("%d",&n);

    printf("Gross salary is %.2f", (n*1.75));

}
```

Input

Enter basic salary: 100000

Output

Gross salary is 175000.00

Q4. Any year is input through the keyboard. Write a program to determine whether the year is a leap year or not.

```
#include <stdio.h>

int main() {
    int year;

    printf("Enter a year: ");
    scanf("%d", &year);

    if (year % 400 == 0) {
        printf("%d is a leap year.\n", year);
    } else if (year % 100 == 0) {
        printf("%d is not a leap year.\n", year);
    } else if (year % 4 == 0) {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }

    return 0;
}
```

Input

Enter a year: 2023

Output

2023 is not a leap year

Q5. Given 3 points (x1,y1),(x2,y2) and (x3,y3) ,write a program to check if all the three points fall on one straight line.

```
#include <stdio.h>

int main()
```

```

{
float x1, y1, x2, y2, x3, y3, m, n;

printf ("Enter points (x1, y1)\n");

scanf ("%f%f", &x1, &y1);

printf ("Enter points (x2, y2)\n");

scanf ("%f%f", &x2, &y2);

printf ("Enter points (x3, y3)\n");

scanf ("%f%f", &x3, &y3);
m = (y2 - y1) / (x2 - x1);
n = (y3 - y2) / (x3 - x2);

if( m == n)
printf ("All 3 points lie on the same line\n");

else
printf ("All 3 points do not lie on the same line\n"); return 0;
}

```

Input

Enter points (x1, y1)

(1.0, 2.0)

Enter points (x2, y2)

(2.0, 5.0)

Enter points (x3, y3)

(10.0, 15.0)

Output

All 3 points do not lie on the same line

Q6. Given the coordinates (x, y) of a center of a circle and its radius, write a C program which will determine whether a point lies inside the circle, on the circle or outside the circle. (Hint: #include <math.h>. Use sqrt() and pow() functions)

```

#include <stdio.h>
#include <math.h>
int main() {
float center_x, center_y, radius;
printf ("Enter center coordinates (x, y): ");
scanf ("%f, %f", &center_x, &center_y);
printf ("Enter circle radius: ");
scanf ("%f", &radius);
float point_x, point_y;
printf ("Enter point coordinates (x, y): ");
scanf ("%f, %f", &point_x, &point_y);

```


5

```

float distance = sqrt(pow(point_x - center_x, 2) + pow(point_y - center_y, 2));
if (distance < radius)
printf ("The point is inside the circle.\n");
else if (distance == radius) {
printf ("The point is on the circle's border.\n"); }
else {
printf ("The point is outside the circle.\n"); }
return 0;
}

```

Input

Enter center coordinates (x,y): (1.0, 0.0)

Enter circle radius: 5.0

Enter point coordinates (x,y): (8.0, 0.0)

Output

The point is outside the circle

Q7. Any character is entered through the keyboard, write a C program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol.

```

#include <stdio.h>

int main() {
char ch;

printf ("Enter a character: ");

scanf ("%c", &ch);

if (ch >= 'A' && ch <= 'Z')
printf ("%c is a capital letter.\n", ch);

else if (ch >= 'a' && ch <= 'z')
printf ("%c is a small case letter.\n", ch);

else if (ch >= '0' && ch <= '9')

printf ("%c is a digit.\n", ch);

else
printf ("%c is a special symbol.\n", ch);

return 0;
}

```


Input

Enter a character: 7

Output

7 is a digit.

Q8. Given as input an integer number of seconds, write a program to print as output the equivalent time in hours, minutes and seconds. Recommended output format is something like 7322 equivalent to 2 hours 2 minutes 2 seconds.

```
#include <stdio.h>
```

```
int main() {
```

```
    int seconds, hours, minutes, sec1;
```

```
    printf("Enter the number of seconds: ");
```

```
    scanf("%d", &seconds);
```

```
    hours = seconds / 3600;
```

```
    minutes = (seconds % 3600) / 60;
```

```
    sec1 = seconds % 60;
```

```
    printf("%d seconds is equivalent to %d hours %d minutes %d seconds\n", seconds, hours, minutes, sec1);
```

```
    return 0;
```

```
}
```

Input

Enter the number of seconds: 7323

Output

7323 seconds is equivalent to 2 hours 2 minutes
3 seconds

Flowchart and Algorithm

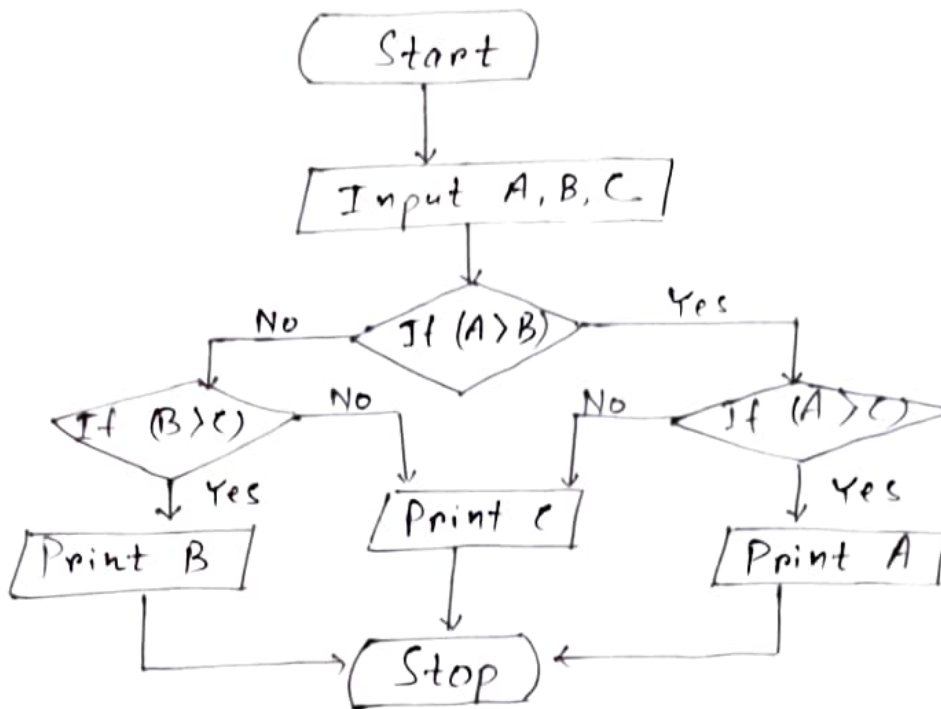
Q1. Write an algorithm to determine the maximum of three numbers. Also Draw the corresponding flowchart.

Algorithm

3 Numbers (a ,b, c)

$\text{max} = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);$

Flowchart



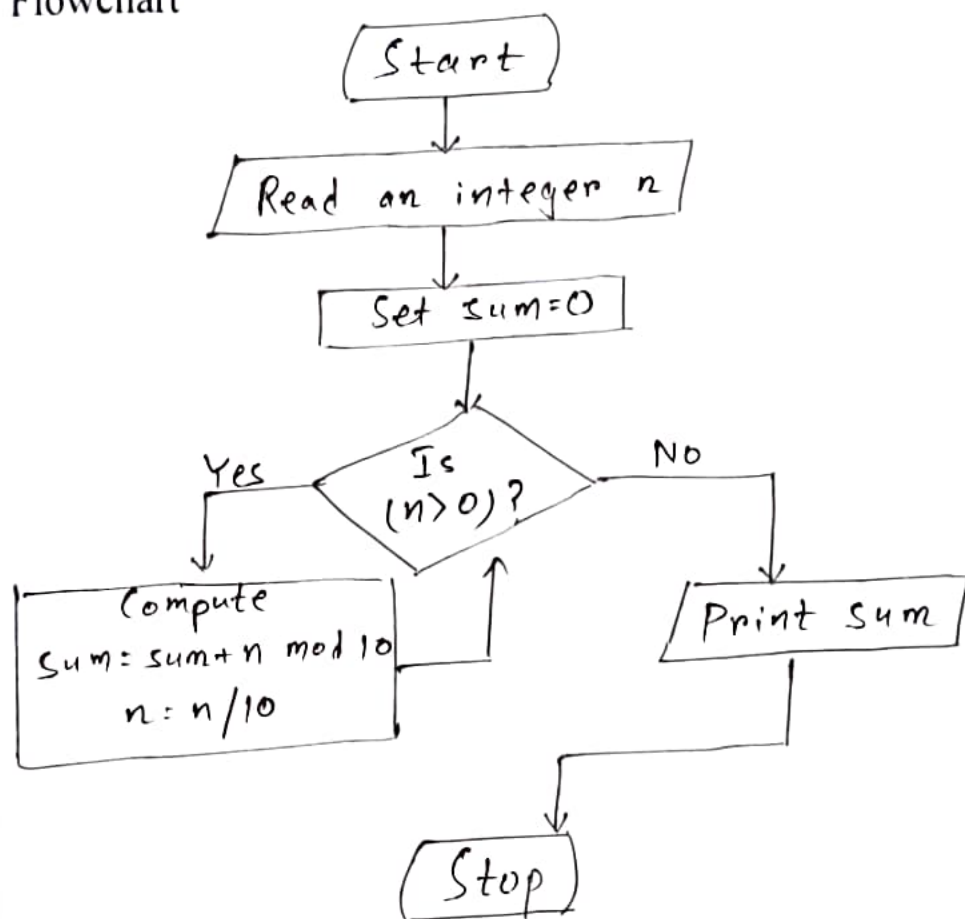
9

Q2. Write an algorithm to determine the sum of individual digits of a given integer. Also draw the corresponding flowchart.

Algorithm

1. num, sum=0
2. sum=sum+(num%10)
3. num=num/10
4. if (num>0) goto 2.

Flowchart



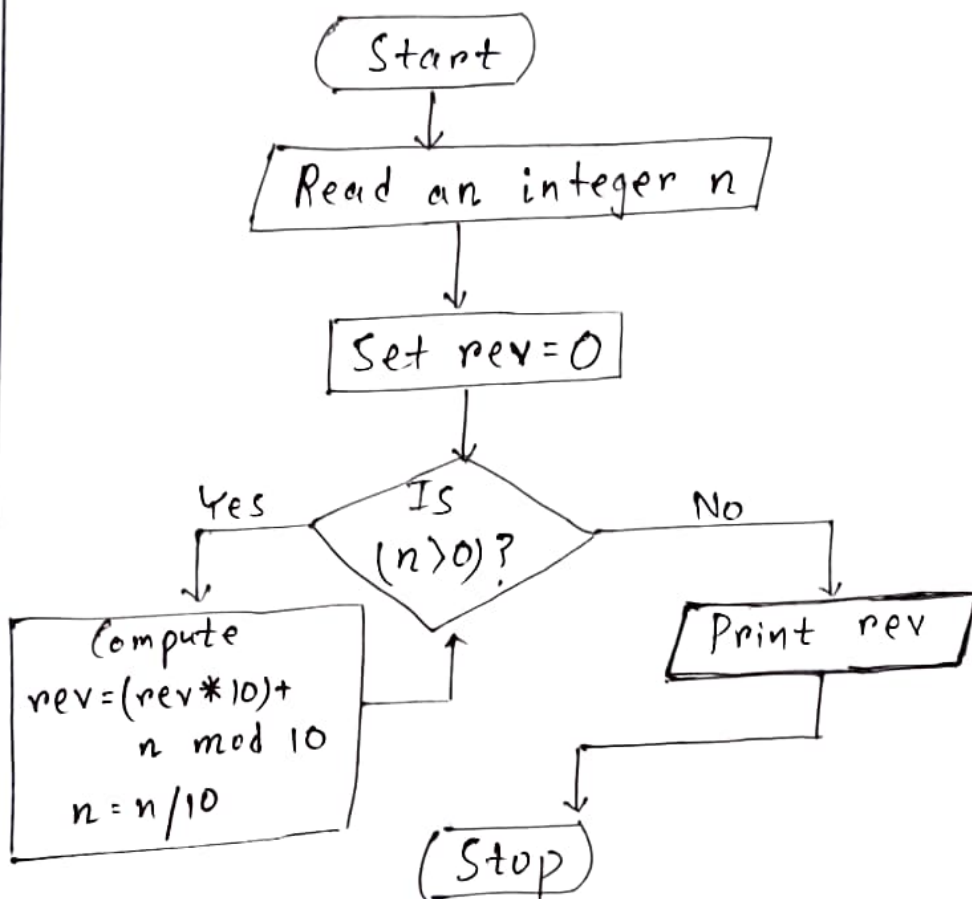
(7)

Q3. Write an algorithm to print the reverse of a number read as input. Also draw the corresponding flow chart.

Algorithm

1. num, rev=0
2. $rev = (rev * 10) + (num \% 10)$
3. $num = num / 10$
4. if ($num > 0$) goto 2.

Flowchart

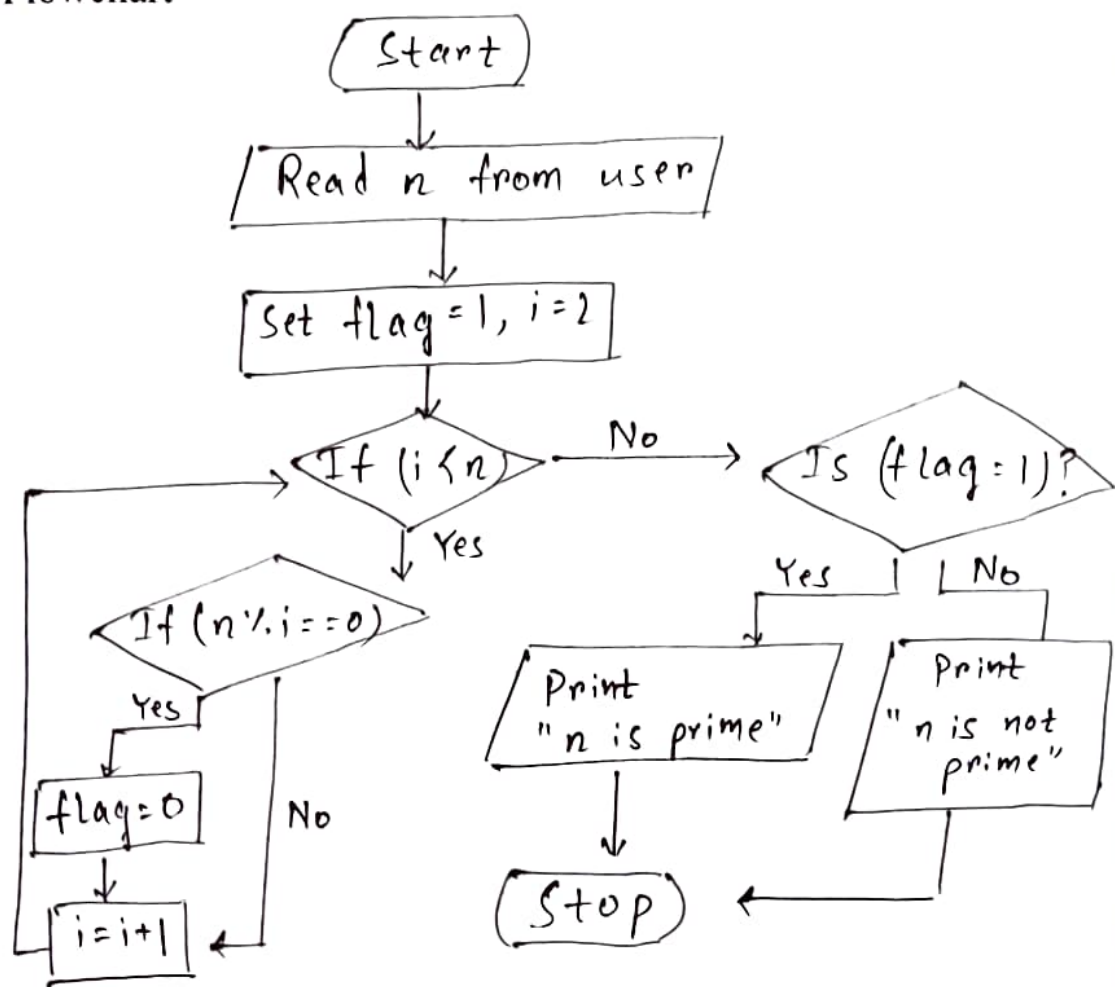


Q4. Write an algorithm to determine whether a given number is prime or not. Also draw the corresponding flowchart.

Algorithm

1. Read the number N
2. Start the loop from $i = 2$ to $i = N - 1$
3. If N is divided by " i " in the loop then break the loop
4. If $(i == N)$ that means loop has executed till the end of range so N is not divided by any number So it is a prime number
5. Else it is not a prime number

Flowchart



Loops

Q1. Write a C program which accepts a number n and prints

- a) All integers divisible by n between 1 and 100 where value of n is provided by the user
- b) All prime numbers between 1 and n
- c) All prime factors of n
- d) Octal equivalent of n
- e) Sum of digits
- f) Factorial of n
- g) Reverse of n

```
#include <stdio.h>
```

```
void 1a(int n)
{
    for(int i=1; i<=100; i++)
```

```
    if (i%n==0)
```

```
        printf("%d\n",i);
```

```
}
```

```
int isp(int n)
```

//Function to check whether n is prime or not

```
{
    for (int j=2; j<n; j++)
```

```
    { if(n%j==0)
```

```
        return 0;
```

```
    }
```

```
    return 1;
```

```
}
```

```
void 1b(int n)
```

```
{
```

```
    for(int i=2; i<n; i++)
```

```
        if(isp(i))
```

```
            printf("%d\n",i);
```

```
}
```

```
void 1c(int n)
```

```
{
```

```
    for(int i=2; i<n; i++)
```

```
display(&&sum);
```

```
printf("\n");
}
```

```
void test10() {
```

```
int octalSum(100); // 0
```

```
while (n != 0) {
```

```
octalSum += n % 8; n = n / 8; // 0
```

```
}
printf("Octal Equivalent:"); for (int i = 0; i <= 0; i++)
```

```
printf("\n octalSum");
```

```
}
void test11() {
```

```
int sum=0;
```

```
while(n!=0)
```

```
{
sum+=n%10;
```

```
n = n/10;
```

```
printf("Sum=%d", sum);
```

```
void test12() {
```

```
int n;
```

```
printf("Enter Number:");
```

```
scanf("%d",
```

```
printf("Reverse=%d", n);
```

```
void test13() {
```

```
int n;
```

```
while(n!=0)
```

```
{
n = n%10;
```

```
n = n/10;
```

```
printf("Reverse=%d", n);
```

```
void main() {
```

```
int n;
```

```
printf("Enter the Number: ");
```

```
scanf("%d", &n);
```


1e(n):

//Checking the user's choice and changing the function called accordingly

}

Input phase

Enter the number: 23

Output phase

(a) 23
46
69
92

(b) 2
3
5
7
11
13
17
19

c) No output
(23 is a prime no.)

d) Octal equivalent = 27

e) Sum = 5

f) Factorial = 862453760

g) Reverse = 32

Q2. Write a C program to find out the sum of the following series:

a) $S = 1 + 2 + 3 + 4 + \dots n$

b) $S = 1.2 + 2.3 + 3.4 + 4.5 + \dots n(n+1)$

c) $S = 1! + 2! + 3! + \dots n!$

d) $S = 1@ + 2@ + 3@ + \dots n@$

where $n@$ is the sum of all factors of n . For e.g.,

$6@ = 1 + 2 + 3 + 6 = 12$

```
#include <stdio.h>

void 2a(int n)
{
    printf("Sum=%d", ((n*(n+1))/2));
}

void 2b(int n) {
    printf("Sum=%d", ((n*(n+1)*(n+2))/3));
}

void 2c(int n)
{
    int sum=0, f=1;

    for(int i=1; i<=n; i++){
        sum+=f;
        f=f*(i+1);
    }    printf ("Sum=%d", sum);
}

void 2d(int n) {
    int sum=0;

    for(int i=1; i<=n; i++) {

        int sum1=1;

        for(int j=2; j<=i; j++) {
            if(i%j==0) sum1+=j;
        }
    }
}
```

```
sum+=sum1, )  
printf("Sum=%d", sum);  
}  
  
void main()  
{  
    int n;  
    printf("Enter the Number :");  
  
    scanf("%d", &n);  
  
    2e(n);           //Checking the user's choice and changing the function called accordingly  
}
```

Input

Enter the number: 5

Output

- a) Sum = 15
- b) Sum = 70
- c) Sum = 153
- d) Sum = 21

Q3. Write a program to generate all combinations of digit 1, 2 and 3 using a for loop.

```
#include <stdio.h>

int main() {
    int i, j, k;

    for(i = 1; i <= 3; i++) {
        for(j = 1; j <= 3; j++) {
            for(k = 1; k <= 3; k++) {
                if (i != j && i != k && j != k)
                    printf("%d %d %d\n", i, j, k);
            }
        }
    }

    return 0;
}
```

Output		
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

Q4. Write a C program to print the first n numbers of the fibonacci sequence. (0,1,1,2,3,5,8,...)

```
#include <stdio.h>

void main() {
    int a=0, b=1, c, n;
    printf("Enter the number");
    scanf("%d", &n);
    printf("The fibonacci series is:%d", a);
    for (int i=1; i<n; i++) {
        c = a+b;
        a=b;
        b=c;
        printf("%d", c);
    }
}
```

Input: Enter the number: 5
Output: The fibonacci series is: 0, 1, 1, 2, 3

Q5. Write a C program which prints the first 10 happy numbers if you iterate the process (assume maximum 100 iterations) of summing the squares of the decimal digits of a number and if the process terminates in 1, then the original number is called a Happy number. For example: 7 is a happy number ($7 \rightarrow 49 \rightarrow 97 \rightarrow 130 \rightarrow 10 \rightarrow 1$)

```
#include <stdio.h>
```

```
int finding_sum_of_squares
```

```
{
    int sum=0;
```

```
while(n>0){
```

```
int d=n%10; sum+=d*d; n/=10;
```

```
} return sum;
```

```
}
```

```
void main()
```

```
{
    int i,j=0;
```

```
for(i=1; i<=10; i++){
```

```
int sum;
```

```
sum=finding_sum_of_squares(i);
```

```
if(sum==1){
```

```
printf("%d\n",i); break;
```

```
}
```

```
}
```

```
}
```

Output

1

10

13

19

23

28

31

32

44

68

88

12

Q6. Write a C program that prints the following patterns, taking the input n from the user

a) 1	b) 1	c) 1	d) 43210
12	12	121	321
123	123	12321	2
1234	1234	1234321	

e) 1
 121
 12321
 1234321
 12321
 121
 1

```
#include <stdio.h>
```

```
void 6a(int n)
```

```
{
for(int i=1; i<=n ; i++) {
for(int j=1; j<=i; j++)
printf("%d",j);
printf("\n");
}
}
```

```
void 6b(int n)
```

```
{
for (int i=1; i<=n ; i++) {
for (int j=1; j<=n-i; j++)
printf(" ");
}
```

for all $n \in \mathbb{N}$, $p_n \in \mathbb{R}$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and

and $\sum_{n=1}^{\infty} p_n < \infty$

and $\sum_{n=1}^{\infty} p_n < \infty$


```

printf(" ");
for(j=1;j<=i;j++)
printf("%d",j);
for(j=i-1;j>=1;j--)
printf("%d",j);
printf("\n");
} for(i=n-1;i>=1;i--) {
for(j=i;j<n;j++) printf(" ");
for(j=1;j<=i;j++) printf("%d",j);
for(j=i-1;j>=1;j--)
printf("%d",j);
printf("\n"); }
}

void main()
{
int n;
printf ("Enter the Number :");      scanf ("%d", &n);

6d(n);                             //Checking the user's choice and changing the function called accordingly
}

```

Input

Enter the number : 5

Output

a) 1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

b) 1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

c) 1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

d) 5 4 3 2 1 0
4 3 2 1
3 2

e) 1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1

Array

Q1. Write a program in C to reverse the contents of the elements of an integer array.

```
#include <stdio.h>

void reverse_array(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        int temp = arr[i], rev=0;
        while(temp>0)
        { rev=rev*10+(temp%10); temp=temp/10; }
        arr[i]=rev;
    }
}

int main() {
    int arr[] = {19, 32, 32, 44, 56}; int n = sizeof(arr) / sizeof(arr[0]);

    printf("Original array: %s", Arrays.toString(arr));

    reverse_array(arr, n);

    printf("Reversed array: %s", Arrays.toString(arr));

    return 0;
}
```

Output

Original array: [19, 32, 32, 44, 56]
 Reversed array: [56, 44, 32, 32, 19]

Q2. Write a menu-driven program for accepting values in two square matrices of 3x3 dimension and generate their sum, difference and product.

```
#include <stdio.h>
```

```
int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i, j, p[100][100], d[100][100];
    printf("Enter the number of rows (between 1 and 100): ");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and 100): ");
    scanf("%d", &c);
    printf("Enter elements of 1st matrix\n");
    for (i = 0; i < r; ++i) {
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
        printf("Enter elements of 2nd matrix\n");
        for (i = 0; i < r; ++i) {
            for (j = 0; j < c; ++j) {
                printf("Enter element b%d%d: ", i + 1, j + 1);
                scanf("%d", &b[i][j]);
            }
        }
        for (i = 0; i < r; ++i) {
            for (j = 0; j < c; ++j) {
                sum[i][j] = a[i][j] + b[i][j];
                d[i][j] = a[i][j] - b[i][j];
            }
        }
        printf("\nSum of two matrices: \n");
        for (i = 0; i < r; ++i) {
            for (j = 0; j < c; ++j) {
                printf("%d ", sum[i][j]);
                if (j == c - 1) printf("\n\n");
            }
        }
        printf("\nDifference of two matrices: \n");
        for (i = 0; i < r; ++i) {
            for (j = 0; j < c; ++j) {
                printf("%d ", d[i][j]);
                if (j == c - 1) printf("\n\n");
            }
        }
        printf("\nProduct of two matrices: \n");
        for (int i = 0; i < r; ++i) {
            for (int j = 0; j < c; ++j) {
                p[i][j] = 0;
                for (int k = 0; k < 3; ++k) {
                    p[i][j] += a[i][k] * b[k][j]; // Calculate element using summation
                }
            }
        }
        for (i = 0; i < r; ++i) {
            for (j = 0; j < c; ++j) {
                printf("%d ", p[i][j]);
                if (j == c - 1) printf("\n\n");
            }
        }
        return 0;
    }
```

Input

Enter the number of rows (between 1 and 100): 2

Enter the number of columns (between 1 and 100): 2

Enter elements of 1st matrix:

Enter element a₁₁: 1

Enter element a₁₂: 2

Enter element a₂₁: 3

Enter element a₂₂: 4

Enter elements of 2nd matrix;

Enter element b11: 1

Enter element b12: 2

Enter element b21: 3

Enter element b22: 4

Output

Sum of two matrices:

2 4

6 8

Difference of two matrices:

0 0

0 0

Product of two matrices:

7 10

15 22

Q3. Write a program to find the range of a set of integers entered by the user. Range is the difference between the smallest and biggest number in the list.

```
#include <stdio.h>
void main()
{
    int n,min=999999,max=-9999999;

    printf("Enter no. of elements");          scanf("%d",&n);

    int a[n];
    printf("Enter array elements");

    for(int i=0;i<n;i++){

        scanf("%d",&a[i]);
        if(a[i]>max)
            max=a[i];
        if(a[i]<min)
            min=a[i];
    }
    printf("Range is %d",(max-min));
```

Input

Enter no. of elements: 5

Enter array elements

{1, 2, 3, 4, 5}

Output

Range is 4

Q4. Write a C program which accepts ten integers from the user and prints them in ascending order. Use an array to store the integers.

```
#include <stdio.h>
```

```
int main() {
    int numbers[10];
```

```

printf("Enter ten integers:\n");
for (int i = 0; i < 10; i++) {

    printf("Enter integer # %d: ", i + 1);           scanf("%d", &numbers[i]);

}

for (int i = 0; i < 9; i++) {
    for (int j = 0; j < 9 - i; j++) {
        if (numbers[j] > numbers[j + 1]) {

            int temp = numbers[j];

            numbers[j] = numbers[j + 1];

            numbers[j + 1] = temp; }

    }

printf("\nSorted numbers in ascending order:\n");

for (int i = 0; i < 10; i++) {
    printf("%d ", numbers[i]);
}

printf("\n");

return 0; }

```

Input

Enter ten integers:

↙ 2, 5, 1, 4, 7, 9, 6, 10, 12, 11 ↘

Output

Sorted numbers in
ascending order

↙ 1, 2, 4, 5, 6, 7, 9, 10, 11, 12 ↘

Q5. Write a C program to replace a square matrix by its transpose without using a second matrix.

```
#include <stdio.h>
```

```
int main() {
    int rows, columns;
```

```
    printf("Enter the number of rows: ");           scanf("%d", &rows);
    printf("Enter the number of columns: ");         scanf("%d", &columns);
```

```
    int matrix[rows][columns];
```

```
    for (int i = 0; i < rows; i++) {
```

```

for (int j = 0; j < columns; j++) {
    printf("Enter element at row %d, column %d ", i + 1, j + 1);
    scanf("%d", &matrix[i][j]);
}

printf("\nTranspose of the matrix:\n");

for (int i = 0; i < columns; i++) {
    for (int j = 0; j < rows; j++) {
        printf("%d ", matrix[j][i]);
    }
    printf("\n");
}

return 0;
}

```

Original matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Transposed Matrix

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Q6. Write a program which takes some numbers and computes the standard deviation of them.

Formulas: For a set of n values; $x_1, x_2, x_3, \dots, x_n$ the average or mean is given by

$$\bar{x} = (\sum x_i) / n$$

The standard deviation is given by: $s = \sqrt{(\sum ((x_i - \bar{x}))^2) / (n-1)}$

```

#include <stdio.h>
#include <math.h>
int main()

```


22

```

{
int n=0,i,sum=0;

printf("Enter no. of elements:");          scanf("%d",&n);

int a[n];
printf ("Enter array elements:");

for(i=0; i<n; i++)
{scanf ("%d",&a[i]);
sum+=a[i];
}
double avg=(sum*1.0)/n, s1=0;
for(i=0;i<n;i++)

s1+=pow(a[i]-avg,2);

s1/=(n-1);
printf("The s.d. is %lf",sqrt(s1));

}

```

Input

Enter no. of array elements: 3

Enter array elements: 1, 5, 4

Output

The s.d. is 2.081666

Function and Pointer

Q1. Write recursive functions for following tasks:
a) Binary equivalent of a number.

```

#include <stdio.h>
int dtb(int n) {          // function to convert decimal to binary
if(n==0)
return 0;
else return ((n%2)+(10*dtb(n/2)));
}
void main()

```



```
{
int n=14;
printf ("Binary equivalent is %d",dtb(n));
}
```

Input: 7

Output: Binary equivalents is 111

b) Sum of individual digits of a number passed as argument

```
#include <stdio.h>
int sod(int n)
{
if(n==0)
return 0;
else return ((n%10)+sod(n/10));
}
void main()
{
int n=23;
printf ("Sum of digits is %d", sod(n));
}
```

Input: 23

Output: Sum of digits is 5

Q2. Write a C program which accepts a full name from the user prints the initials. E.g. SRT for Sachin Ramesh Tendulkar

```
#include <stdio.h>
#include <string.h>
void main()
{
char n[100];
printf("Enter Full Name:");
fgets(n,100,stdin);
```

28

```
char *t= strtok(n, " ");
while (t!=NULL)
{
printf("%c",t[0]);    t= strtok(NULL, " ");
}
}
```

Input

Enter Full TName: Tathagata Sur

Output TS

Q3. Write a program to count the number of occurrences of any two Vowels in succession in a line of text.

```
#include <stdio.h>

#include <string.h>

int isvow(char c)
{ int i=0;

if(c>='A' && c<='Z') c+=32;

if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')        i=1;

return i;
}

int main() {

char s[ 100 ];
fgets (s,100, stdin);

int j=0;
for(int i=0;i<strlen(s)-1;i++)

{ if (isvow(s[i])==1 && isvow(s[i+1])==1)

j++;

} printf("The number of occurances of vowel double is %d/n",j);

return 0;

}
```

Input

Hello World

Output

The number of occurances of vowel double is 0

Q4. Write a C program which accepts a string from the user and performs the following tasks.

- a. Check whether it is palindrome or not.. [Example of a palindrome string: "abcba", "abba"]
- b. Counts the number of characters and words in it.

```
#include <stdio.h>

#include <string.h>

int ispal(char s[]) {
    int c=1, l=strlen(s);

    for(int i=0; i<l; i++) {

        if(s[i]!=s[l-i-1] || s[i]==(s[l-i-1]+32) || s[i]==(s[l-i-1]-32))
        {c=0;          break;
        }

    }
    return c; }

int main() {

    char s[100];
    printf("Enter the String\n"); gets(s);
    int l=strlen(s);
    char t[l];
    int c=0;

    for(int i=0; i<l; i++)
    {
        char ch=s[i];
        if ((ch>=65 && ch<=90) || (ch>=97 && ch<=122))

            t[c++]=ch;
    }

    t[c]='\0';
    if (ispal(t))

        {printf("Palindrome\n");}
    else {printf("Not Palindrome\n");}
    int w=0;
    c=0;
    for(int i=0; i<l-1; i++)
    {
        if((s[i]>=65 && s[i]<=90)|| (s[i]>=97 && s[i]<=122))
        {
            c++;
            if(s[i+1]==' ')
                w++;
        }
    }
```

```

}
printf("No. of chars=%d and No. of Words=%d", ++c, ++w);
return 0;
}

```

Input abba

Output Palindrome

No. of chars = 4 and No. of words = 1

Q5. Write a C function which accepts a string str1 and returns a new string str2 which is str1 with each word reversed.

```

#include <stdio.h>

#include <string.h>

int main() {
    char s[100];

    printf("Enter the string "); gets(s);
    int l = strlen(s);
    char t[l+1];

    for(int i=0; i<l; i++) {
        t[i] = s[l-i-1];
    }

    t[l] = '\0';
    printf("\n After the reverse of a string: %s ", t); return 0;
}

```

Input Hello World

Output olleH dlrow

Q6. Write a function squeeze(s,c) which removes all occurrences of the character c from the string s.

```

#include <stdio.h>

#include <string.h>

int main() {

```

```

char str[100], ch; int i, len, j;

printf("\n Please Enter any String : ");

gets(str);

printf("\n Please Enter the Character that you want to Remove : ");

scanf("%c", &ch);

len = strlen(str);
for(i = 0; i < len; i++)
{
    if(str[i] == ch)
    {
        for(j = i; j < len; j++)
            str[j] = str[j + 1];

        len--;          i--;
    }
}

printf("\n The Final String after Removing All Occurrences of '%c' = %s ", ch, str);

return 0;

```

Input: Please Enter any String:
Hello world

Please Enter the Character that you want to
remove: o

Output: The final String after Removing all
Occurance of 'o' = Hell world

Files

Q1. Write a C-program which reads any source-code file written in C, and containing comment blocks within it, and then writes the source-code back into an output file, but with all comment lines and blocks removed. Check the size and actual content of the two source code files manually to establish the accuracy of your program.

22

```
#include <stdio.h>
```

```
int main() {
```

```
    FILE *inputFile, *outputFile;
```

```
    char ch;
```

```
    int inCommentBlock = 0;
```

```
    inputFile = fopen("input.c", "r");
```

```
    outputFile = fopen("output.c", "w");
```

```
    if (inputFile == NULL || outputFile == NULL) {
```

```
        printf("Error opening files!\n");
```

```
        return 1;
```

```
    }
```

```
    // Process the input file character by character
```

```
    while ((ch = fgetc(inputFile)) != EOF) {
```

```
        if (inCommentBlock == 0) {
```

```
            // Not in a comment block
```

```
            if (ch == '/') {
```

```
                ch = fgetc(inputFile);
```

```
                if (ch == '/') { // Single-line comment
```

```
                    while (ch != '\n' && ch != EOF) {
```

```
                        ch = fgetc(inputFile);
```

```
                    }
```

```
                } else if (ch == '*') { // Multi-line comment
```

```
                    inCommentBlock = 1;
```

```
                } else {
```

```
                    fputc('/', outputFile); // Not a comment, write the '/' back
```

```
                    fputc(ch, outputFile);
```

```
                }
```

```
            } else {
```

```
                fputc(ch, outputFile); // Not a comment, write the character
```

```
            }
```

```
        } else {
```

```
            // In a multi-line comment block
```

```
            if (ch == '*') {
```

```
                ch = fgetc(inputFile);
```

```
                if (ch == '/') { // End of multi-line comment
```

```
                    inCommentBlock = 0;
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    fclose (inputFile);
```

```
    fclose (outputFile);
```

```
    printf("Comments removed successfully. Check output.c for the modified code.\n");
```

```
    return 0;
```

```
}
```


Q2. Write a C program which accepts sentences in an array of character strings in the main function and then passes each of those strings to an encoder function. The encoder function converts the string using a bit-operation logic - such as shifting by a fixed number of bits, or xor-ing with bit-wise complement value, or any other such suitable techniques. Test it out with a source code file content.

```
#include <stdio.h>
#include <string.h>

char *encoder(char *sentence) {
    char *encoded_sentence = malloc(strlen(sentence) + 1);
    int i;

    for (i = 0; sentence[i] != '\0'; i++) {
        // XOR with bit-wise complement value (example technique)
        encoded_sentence[i] = sentence[i] ^ 255;
    }

    encoded_sentence[i] = '\0';
    return encoded_sentence;
}

int main() {
    char *sentences[] = {
        "This is a sample sentence.",
        "Another sentence for testing.",
        "Source code file content can also be encoded."
    };
    int num_sentences = sizeof(sentences) / sizeof(sentences[0]);
    int i;

    for (i = 0; i < num_sentences; i++) {
        char *encoded = encoder(sentences[i]);
        printf("Original sentence %d: %s\n", i + 1, sentences[i]);
        printf("Encoded sentence %d: %s\n", i + 1, encoded);
        free(encoded);
    }

    return 0;
}
```

Output

Original sentence 1: This is a sample sentence

Encoded sentence 1: 7Y?YH?5?

Original sentence 2: Another sentence for testing

Encoded sentence 2: ?0?+?Y?

Original sentence 3: Source code file content can also be encoded

Encoded sentence 3: ?+? ?H?V?+A?

Numerical Methods

Q1. Generate a table of $\sin(x)$, $\cos(x)$ and $\tan(x)$ for 0 to 360 degree with an interval of 15 using the series expansion as well as the library function accurate upto 5th place of decimal as follows:

$\sin(\text{series})$, $\sin(\text{library})$, $\cos(\text{series})$, $\cos(\text{library})$, $\tan(\text{series})$, $\tan(\text{library})$

```
#include <stdio.h>
#include <math.h>
void main()
{
    int ch;
    printf("Enter 1 for sine series, 2 for cosine series and 3 for tangent series");

    scanf("%d",&ch);
    switch(ch)
    {
        case 1:for(int i=0;i<=360;i+=15)

            printf("Sin of %d=%.5f\n", i, sin(3.14159265359*i/180));

            break;
        case 2:for(int i=0;i<=360;i+=15)

            printf("Cos of %d=%.5f\n", i, cos(3.14159265359*i/180));

            break;
        case 3:for(int i=0;i<=360;i+=15)

            printf("Tan of %d=%.5f\n",i,tan(3.14159265359*i/180));

            break;

        default: printf("Invalid choice");

    }
}
```

Input: Enter 1 for sine series, 2 for cosine series and 3 for tangent series: 1

Output:

sin of 0 = 0.00000	sin of 60: 0.86603
sin of 15 = 0.25882	sin of 75: 0.96593
sin of 30 = 0.50000	sin of 90: 1.00000 ...
sin of 45 = 0.70711	sin of 105: 0.96593
	sin of 120: 0.86603

Q2. Find the value of $\ln(2)$ by the series

$$\ln(1+x) = x - \frac{(x^2)}{2} + \frac{(x^3)}{3} + \dots$$

and then verify the value by the series

$$(e^x) = 1 + x + \frac{(x^2)}{2!} + \frac{(x^3)}{3!} + \dots$$

```
#include <stdio.h>

#include <math.h>

int fact(int n)
{
    if(n==1)
        return 1;
    return n*fact(n-1);
}

void main()
{
    int n, ch;
    printf("Enter the no. :");
    scanf("%d", &n);
    printf("Enter 1 for ln(x) ,2 for e^x:");
    scanf("%d", &ch);

    if (ch==1)
    {
        n--;
        double sum=0.0, c=1.0;

        for(int i=1; i<=100; i++) {
            sum+=((c*pow(n, i))/ i);
            c=c*-1.0;
        }
        printf("%lf", sum);
    }
    else if(ch==2)
    {
        double sum=1.0;
        for(int i=1; i<=10; i++) sum+=(1.0*pow(n,i)/fact(i)); printf("%lf", sum);
    }
    else printf("Invalid choice");
}
```

Output

$$\ln 2 = 0.693147$$