

// Q1. Write a C program for printing prime factors of a number using functions.

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

```
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i==0)
            return 0;
        else if (i+1==x)
            return x;
    }
}
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    printf("Prime Factors: ");
    for (int i=2;i<=n;i++)
    {
        if (n%i==0 && prime_number(i))
            printf("%d ",i);
    }
    return 0;
}
```

Output:

Enter Number: 2310

Prime Factors: 2 3 5 7 11

// Q2. Write a C program for printing prime numbers between a interval.

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

```
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i == 0)
            return 0;
        else if (i+1 == x)
            return x;
    }
}
```

```
int main()
{
    int a,b;
    printf("Enter Starting Number: ");
    scanf("%d",&a);
    printf("Enter Ending Number: ");
    scanf("%d",&b);
    printf("Prime Numbers between %d and %d: ",a,b);
    if (a<2)
        a=2;
    for (int i=a;i<=b;i++)
    {
        int c=prime_number(i);
        if (c!=0)
            printf("%d ",i);
    }
    return 0;
}
```

Output:

Enter Starting Number: 6

Enter Ending Number: 24

Prime Numbers between 6 and 24: 7 11 13 17 19 23

// Q3. Write a C program to convert decimal to binary numbers & vice versa.

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

```
#include <math.h>
```

```
int Decimal_to_Binary(int x)
```

```
{
    int A[64],i=0;
    printf("Decimal: %d --> Binary: ",x);
    if (x == 0)
        printf("0");
    while (x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
```

```
int Binary_to_Decimal(int x)
```

```
{
    int A[128],i=0,s=0;
    printf("Binary: %d --> Decimal: ",x);
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i] == 1)
            s+=A[i]*pow(2,i);
        i++;
    }
    printf("%d",s);
}
```

```
int main()
```

```
{
    int n,i;
    printf("Enter\n1. for Decimal To Binary, or\n2. for Binary to Decimal: ");
    scanf("%d",&n);
    switch(n)
    {
        case 1:
        {
            printf("Enter Decimal Number: ");
            scanf("%d",&i);
            Decimal_to_Binary(i);
            break;
        }
    }
}
```

```

        case 2:
        {
            printf("Enter Binary Number (Only 0 and 1): ");
            scanf("%d",&i);
            Binary_to_Decimal(i);
            break;
        }
        default:
        {
            printf("ERROR!! Enter from above choices.");
        }
    }
    return 0;
}

```

Output:

Enter

1. for Decimal To Binary, or

2. for Binary to Decimal: 1

Enter Decimal Number: 13

Decimal: 13 --> Binary: 1101

Enter

1. for Decimal To Binary, or

2. for Binary to Decimal: 2

Enter Binary Number (Only 0 and 1): 1100110

Binary: 1100110 --> Decimal: 102

// Q4. Write a C program to convert octal to binary & vice versa.

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

```
#include <math.h>
```

```
void Decimal_to_Binary(int x)
```

```
{
    int A[64],i=0;
    while(x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
```

```
int Binary_to_Decimal(int x)
```

```
{
    int A[64],i=0,s=0;
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i]==1)
            s+=A[i]*pow(2,i);
        i++;
    }
    return s;
}
```

```
int main()
```

```
{
    int c,n;
    printf("Enter\n1.for Octal to Binary, or\n2. for Binary to Octal: ");
    scanf("%d",&c);
    switch(c)
    {
        case 1:
        {
            char A[16];
            int d;
            printf("Enter Octal Number: ");
            scanf("%s",A);
            sscanf(A,"%o",&n);
            printf("Octal: %s --> Binary: ",A);
            Decimal_to_Binary(n);
            break;
        }
        case 2:
```

```

    {
        char A[16];
        printf("Enter Binary Number (only 0 and 1): ");
        scanf("%d",&n);
        int d=Binary_to_Decimal(n);
        sprintf(A,"%o",d);
        printf("Binary: %d --> Octal: %s",n,A);
        break;
    }
}

```

Output:

Enter

1.for Octal to Binary, or

2. for Binary to Octal: 1

Enter Octal Number: 57

Octal: 57 --> Binary: 101111

Enter

1.for Octal to Binary, or

2. for Binary to Octal: 2

Enter Binary Number (only 0 and 1): 111011

Binary: 111011 --> Octal: 73

```
// Q5. Write a C program to find mean, median, mode.  
#include <stdio.h>
```

```
float mean(float A[], int x)  
{  
    float s = 0;  
    for (int i = 0; i < x; i++)  
        s += A[i];  
    return s / x;  
}
```

```
float median(float A[], int x)  
{  
    if (x % 2 == 0)  
        return (A[x / 2] + A[x / 2 - 1]) / 2;  
    else  
        return A[x / 2];  
}
```

```
void sort_array_increasing(float A[], int x)  
{  
    float t;  
    for (int i = 0; i < x; i++)  
    {  
        for (int j = i + 1; j < x; j++)  
        {  
            if (A[i] > A[j])  
            {  
                t = A[i];  
                A[i] = A[j];  
                A[j] = t;  
            }  
        }  
    }  
}
```

```
float mode(float A[], int x)  
{  
    int max_count = 0, count;  
    float mode_value = A[0];  
    for (int i = 0; i < x; i++)  
    {  
        count = 1; // Count occurrences of A[i]  
        for (int j = i + 1; j < x; j++)  
        {  
            if (A[i] == A[j])  
                count++;  
        }  
  
        if (count > max_count)  
        {
```

```

        max_count = count;
        mode_value = A[i];
    }
}
if (max_count == 1)
{
    printf("No mode exists (all values are unique).\n");
    return -1;
}

return mode_value;
}

int main()
{
    int n;
    printf("Enter Number of Elements: ");
    scanf("%d", &n);
    float Element[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter Element %d: ", i + 1);
        scanf("%f", &Element[i]);
    }
    sort_array_increasing(Element, n);
    printf("Elements in increasing order are: ");
    for (int i = 0; i < n; i++)
        printf("%.2f ", Element[i]);
    printf("\n");
    printf("Mean: %.2f\n", mean(Element, n));
    printf("Median: %.2f\n", median(Element, n));
    float mode_result = mode(Element, n);
    if (mode_result != -1)
        printf("Mode: %.2f\n", mode_result);
    return 0;
}

```

Output:

```

Enter Number of Elements: 5
Enter Element 1: 12
Enter Element 2: 35
Enter Element 3: 15
Enter Element 4: 13
Enter Element 5: 35
Elements in increasing order are: 12.00 13.00 15.00 35.00 35.00
Mean: 22.00
Median: 15.00
Mode: 35.00

```


// Q6. Write a program in C to make such a pattern like a right angle triangle with a number, which will repeat a number in a row.

```
#include <stdio.h>

int main()
{
    int r;
    printf("Enter Rows: ");
    scanf("%d",&r);
    for (int i=1;i<=r;i++)
    {
        for (int j=1;j<=i;j++)
            printf("%d",i);
        printf("\n");
    }
}
```

Output:

Enter Rows: 7

1

22

333

4444

55555

666666

7777777

// Q7 Write a program in C to make a pyramid pattern with numbers increased by 1.

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

```
int main()
{
    int r;
    printf("Enter Rows: ");
    scanf("%d",&r);
    int a=r*(r+1)/2;
    int A[a];
    for (int i=0;i<a;i++)
        A[i]=i+1;
    int t=0;
    for (int i=0;i<r;i++)
    {
        for (int j=0;j<=i;j++)
        {
            printf("%d ",A[t]);
            t++;
        }
        printf("\n");
    }
}
```

Output:

Enter Rows: 4

1

2 3

4 5 6

7 8 9 10

// Q8. Write a program in C to print Floyd's Triangle.

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

```
int main()
{
    int r;
    printf("Enter number of rows: ");
    scanf("%d",&r);
    for (int i=1;i<=r;i++)
    {
        int s=i%2;
        for (int j=1;j<=i;j++)
        {
            printf("%d ",s);
            s=1-s;
        }
        printf("\n");
    }
}
```

Output:

Enter number of rows: 6

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

// Q9. Write a program in C to display a pattern like a diamond.

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

```
int main()
{
    int rows;

    printf("Enter the number of rows (odd number): ");
    scanf("%d",&rows);
    int mid=(rows+1)/2;
    for (int i=1;i<=mid;i++)
    {
        for (int j=1;j<= mid-i;j++)
            printf(" ");
        for (int j= 1;j<=2*i-1;j++)
            printf("*");
        printf("\n");
    }
    for (int i= mid-1;i>=1;i--)
    {
        for (int j=1;j<=mid-i;j++)
            printf(" ");
        for (int j=1;j<=2*i-1;j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

Output:

Enter the number of rows (odd number): 7

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

// Q10. Write a program in C to display a pattern like a diamond.

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

```
void printPascalsTriangle(int rows)
```

```
{
    int arr[rows][rows];
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j <= i; j++)
        {
            if (j == 0 || j == i)
                arr[i][j] = 1;
            else
                arr[i][j] = arr[i - 1][j - 1] + arr[i - 1][j];
        }
    }
    for (int i = 0; i < rows; i++)
    {
        for (int space = 0; space < rows - i - 1; space++)
            printf(" ");
        for (int j = 0; j <= i; j++)
            printf("%d ", arr[i][j]);
        printf("\n");
    }
}
```

```
int main() {
    int n;
    printf("Enter the number of rows for Pascal's Triangle: ");
    scanf("%d", &n);
    if (n <= 0)
        printf("Number of rows must be a positive integer.\n");
    else
        printPascalsTriangle(n);
    return 0;
}
```

Output:

Enter the number of rows for Pascal's Triangle: 5

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

// Q11. Write a program in C to check whether a number can be expressed as the sum of two prime numbers

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

```
int prime_number(int x)
{
    for (int i=2;i<x;i++)
    {
        if (x%i == 0)
            return 0;
        else if (i+1 == x)
            return x;
    }
}
```

```
int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    int a =n/2;
    for (int i=2;i<=a;i++)
    {
        if (prime_number(i) && prime_number(n-i))
            printf("%d = %d + %d\n",n,i,n-i);
    }
    return 0;
}
```

Output:

Enter Number: 100

100 = 3 + 97

100 = 11 + 89

100 = 17 + 83

100 = 29 + 71

100 = 41 + 59

100 = 47 + 53

// Q12. Write a C program to check whether a number is a Strong Number or not.

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

int length(int x)
{
    int len =0;
    while (x>0)
    {
        x=x/10;
        len++;
    }
    return len;
}

void Armstrong_Number(int x)
{
    int l=length(x);
    int A[l],s=0,a=x;
    while (a>0)
    {
        int r=a%10;
        s=s+round(pow(r,l));
        a/=10;
    }
    if (s==x)
        printf("Number %d is an Armstrong Number.\n",x);
    else
        printf("Number %d is NOT an Armstrong Number.\n",x);
}

int main()
{
    int n;
    printf("Enter Number: ");
    scanf("%d",&n);
    Armstrong_Number(n);
    return 0;
}
```

Output:

Enter Number: 9474

Number 9474 is an Armstrong Number.

// Q13. Write C program to convert binary to hexadecimal & vice versa

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

```
#include <math.h>
```

```
int Binary_to_Decimal(int x)
```

```
{
    int A[32],i=0,s=0;
    while (x>0)
    {
        A[i]=x%10;
        x/=10;
        if (A[i]==1)
            s+=A[i]*pow(2,i);
        i++;
    }
    return s;
}
```

```
int Decimal_to_Binary(int x)
```

```
{
    int A[64],i=0;
    if (x==0)
        return 0;
    while (x>0)
    {
        A[i]=x%2;
        x/=2;
        i++;
    }
    for (int j=i-1;j>=0;j--)
        printf("%d",A[j]);
}
```

```
int main()
```

```
{
    int c,n;
    printf("Enter\n1. for Binary to Hexadecimal, or\n2. Hexadecimal to Binary: ");
    scanf("%d",&c);
    switch(c)
    {
        case 1:
        {
            printf("Enter Binary Number: ");
            scanf("%d",&n);
            int d=Binary_to_Decimal(n);
            printf("Binary: %d --> Hexadecimal: %X\n",d,d);
            break;
        }
        case 2:
```



```

    {
        char A[16],B[32];
        printf("Enter Hexadecimal Number: ");
        scanf("%s",&A);
        sscanf(A,"%X",&n);
        printf("Hexadecimal: %s --> Binary: ",A);
        Decimal_to_Binary(n);
        break;
    }
}

```

Output:

Enter

1. for Binary to Hexadecimal, or
2. Hexadecimal to Binary: 1

Enter Binary Number: 101100

Binary: 44 --> Hexadecimal: 2C

Enter

1. for Binary to Hexadecimal, or
2. Hexadecimal to Binary: 2

Enter Hexadecimal Number: 4A

Hexadecimal: 4A --> Binary: 1001010

// Q14. Write a program in C to find the number and sum of all integers between 100 and 200 which are divisible by 9.

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

```
int main()
{
    int s=0;
    printf("Number divisible by 9 between 100 to 200 are: \n");
    for (int i=100;i<=200;i++)
    {
        if (i%9 == 0)
        {
            printf("%d ",i);
            s+=i;
        }
    }
    printf("\nSum: %d",s);
}
```

Output:

Number divisible by 9 between 100 to 200 are:

108 117 126 135 144 153 162 171 180 189 198

Sum: 1683

// Q15. Write a program to find out the second largest factor of a number. Eg.
Input->100, Output-> 50.

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

```
int main()
{
    int n,index=0;
    printf("Enter Number: ");
    scanf("%d",&n);
    int A[n/2];
    for (int i=1;i<=n;i++)
    {
        if (n%i == 0)
        {
            A[index]=i;
            index++;
        }
    }
    printf("Second largest Factor of %d is %d.",n,A[index-2]);
}
```

Output:

Enter Number: 68

Second largest Factor of 68 is 34.

// Q16. Write C Program to Find the Largest Element in an Array.

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

```
int main()
{
    int n;
    printf("Enter Size of an Array: ");
    scanf("%d",&n);
    int A[n];
    for (int i=0;i<n;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    int t;
    for (int i=0;i<n;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (A[i] < A[j])
            {
                t=A[i];
                A[i]=A[j];
                A[j]=t;
            }
        }
    }
    printf("The Largest Element in Array is %d.",A[0]);
}
```

Output:

Enter Size of an Array: 5

Enter Element 1: 1

Enter Element 2: 3

Enter Element 3: 6

Enter Element 4: -8

Enter Element 5: 5

The Largest Element in Array is 6.

// Q17. Write a C Program to Find the Maximum and Minimum in an Array.

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

```
int main()
{
    int n;
    printf("Enter Size of an Array: ");
    scanf("%d",&n);
    int A[n];
    for (int i=0;i<n;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    int t;
    for (int i=0;i<n;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (A[i] < A[j])
            {
                t=A[i];
                A[i]=A[j];
                A[j]=t;
            }
        }
    }
    printf("The Largest Element in an Array is %d.\n",A[0]);
    printf("The Smallest Element in an Array is %d.\n",A[n-1]);
}
```

Output:

Enter Size of an Array: 5

Enter Element 1: 3

Enter Element 2: -5

Enter Element 3: 6

Enter Element 4: -12

Enter Element 5: 4

The Largest Element in an Array is 6.

The Smallest Element in an Array is -12.

// Q18. Write C Program to Search an Element in an Array (Binary search)

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

```
int main()
{
    int a,b,t;
    printf("Enter Size of Array: ");
    scanf("%d",&a);
    int A[a];
    for (int i=0;i<a;i++)
    {
        printf("Enter Element %d: ",i+1);
        scanf("%d",&A[i]);
    }
    printf("Enter Number to search: ");
    scanf("%d",&b);
    int found=0;
    for (int i=0;i<a;i++)
    {
        if (b==A[i])
        {
            printf("Element %d Found at Index %d.\n",b,i);
            found=1;
        }
    }
    if (!found)
        printf("Element %d NOT Found.",b);
}
```

Output:

```
Enter Size of Array: 5
Enter Element 1: 1
Enter Element 2: 6
Enter Element 3: 99
Enter Element 4: 23
Enter Element 5: 42
Enter Number to search: 23
Element 23 Found at Index 3.
```