

EX.NO :1(A)

DATE:

PROGRAMS USING I/O STATEMENTS AND EXPRESSIONS.

/* Program to convert Celsius to Fahrenheit */

Aim:

Algorithm:

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float celsius, fahrenheit;
    clrscr();
    printf("\n Enter the Temperature in Celsius : ");
    scanf("%f",&celsius);
    fahrenheit = (1.8 * celsius) + 32;
    printf("\n Temperature in Fahrenheit : %f ", fahrenheit);
    getch();
}
```

Sample Input and Output:

Input :

Output :

Result:

EX.NO :1(B)

DATE:

/* Program to print the address and size of the data & data type */

Aim:

Algorithm:

Program:

```
#include <stdio.h>
void main()
{
    int x=62;
    char a='b';
    float p=8.5,q=3.32;
// Print Address
    printf("%c is Stored at Address %u\n",a,&a);
    printf("%d is Stored at Address %u\n",x,&x);
    printf("%f is Stored at Address %u\n",p,&p);
    printf("%f is Stored at Address %u\n",q,&q);
// Print Size
    printf("\n size of short: %d", sizeof(short));
    printf("\n size of int: %d", sizeof(int));
    printf("\n size of long int: %d", sizeof(long));
}
```

Sample Input and Output:

Result:

EX.NO :1(C)

DATE:

SIMPLE INTEREST CALCULATION

Aim :

Algorithm:

Program :

```
#include<stdio.h>
int main()
{
    float amount, rate, time, si;
    printf("\nEnter Principal Amount : ");
    scanf("%f", &amount);
    printf("\nEnter Rate of Interest : ");
    scanf("%f", &rate);
    printf("\nEnter Period of Time : ");
    scanf("%f", &time);
    si = (amount * rate * time) / 100;
    printf("\nSimple Interest : %f", si);
    return(0);
}
```

Sample Input and Output:

Input:

Output

Result:

EX.NO:2(A)

DATE:

PROGRAMS USING DECISION-MAKING CONSTRUCTS.

PRINT FIRST N TERMS OF THE FIBONACCI SERIES

Aim:

Algorithm:

Program :

```
#include <stdio.h>
int main()
{
    int a, b, c, i, n;
    /* Reads a number from user */
    printf("Enter value of n to print Fibonacci series : ");
    scanf("%d", &n);
    a = 0;      /* Initialize the value */
    b = 1;
    c = 0;      /* Sum the value */
    for(i=1; i<=n; i++)
    {
        printf("%d, ", c);
```

```
a=b;  
b=c;  
c=a+b; }  
return 0; }
```

Sample input and output

Input :

Output:

Result:

EX.NO :2(B)

DATE:

TO FIND GCD (HCF) OF ANY TWO NUMBERS

Aim :

Algorithm :

Program :

```
#include <stdio.h>
int main()
{
    int i, num1, num2, min, hcf=1;
    /* Reads two numbers from user */
    printf("Enter any two numbers to find HCF: ");
    scanf("%d %d", &num1, &num2);
    min = (num1<num2) ? num1 : num2; // Conditional Operator
    for(i=1; i<=min; i++)
    {
        /* If i is factor of both number */
        if(num1 % i==0 && num2 % i==0)
        {
```

```
        hcf = i;  
    }  
}  
printf("HCF of %d and %d = %d\n", num1, num2, hcf);  
return 0;  
}
```

Sample input and output:

Input :

Output:

Result:

**EX.NO :2(C) TO CHECK WHETHER AN ALPHABET IS VOWEL
OR CONSONANT**

DATE:

Aim :

Algorithm :

Program :

```
#include <stdio.h>
int main()
{
    char ch; // declare a variable to get any character .
    printf("Enter any character: ");
    scanf("%c", &ch); /* Reads a character from user */
    if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' ||
    ch=='O' || ch=='U')
    {
        printf("%c is VOWEL.\n", ch);
    }
}
```

```
else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))  
{  
    printf("%c is CONSONANT.\n", ch);  
}  
return 0;  
}
```

Sample input and output :

Input :

Output:

Result:

EX.NO :2(D)

DATE:

PRINT FLOYD'S TRIANGLE

Aim :

Algorithm :

Program :

```
#include <stdio.h>
int main()
{
    int rows, a, b, number = 1;
    printf("Number of rows of Floyd's triangle to print:");
    scanf("%d",&rows);
    printf("FLOYD'S TRIANGLE\n\n");
    for ( a = 1 ; a <= rows ; a++ )
    {
```

```
for ( b = 1 ; b <= a ; b++ )
{
    printf("%d ", number);
    number++;
}
printf("\n");
return 0;
}
```

Sample input and output

Input :

Number of rows of Floyd's triangle to print:5

Output:

Result:

EX.NO : 3

DATE:

**WRITE A PROGRAM TO FIND WHETHER THE GIVEN YEAR IS
LEAP YEAR OR NOT?**

**(HINT: NOT EVERY CENTURION YEAR IS A LEAP.
FOR EXAMPLE 1700, 1800 AND 1900 IS NOT LEAP YEAR)**

Aim :

Algorithm:

Program :

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int year;
    clrscr();
    printf("Enter the Year (YYYY) : ");
    scanf("%d",&year);
    if(year%4==0 && year%100!=0 || year%400==0)
        printf("\nThe Given year %d is a Leap Year");
```

```
        else
            printf("\nThe Given year %d is Not a Leap Year");
    getch(); }
```

Sample Input and Output :

Input :

Output :

Result:

EX.NO : 4

DATE:

**DESIGN A CALCULATOR TO PERFORM THE OPERATIONS,
NAMELY, ADDITION, SUBTRACTION, MULTIPLICATION,
DIVISION AND SQUARE OF A NUMBER**

Aim:

Algorithm:

Program :

```
#include <stdio.h>
float add(float num1, float num2);
float sub(float num1, float num2);
float mult(float num1, float num2);
float div(float num1, float num2);
float squa(float num1, float num2);
int main()
```

```

{
    char op;
    float num1, num2, result=0.0f;
    printf("WELCOME TO SIMPLE CALCULATOR\n");
    printf("-----\n");
    printf("Enter [number 1] [+ - * /] [number 2]\n");
    /* Input two number and operator from user */
    scanf("%f %c %f", &num1, &op, &num2);
    switch(op)
    {
        case '+':
            result = add(num1, num2);           break;
        case '-':
            result = sub(num1, num2);           break;
        case '*':
            result = mult(num1, num2);          break;
        case '/':
            result = div(num1, num2);           break;
        case 's':
            result = squa(num1, num2);          break;
        default:
            printf("Invalid operator");
    }

    /* Print the result */
    printf("%.2f %c %.2f = %.2f", num1, op, num2, result);
    return 0;
}

/** * Function to add two numbers */
float add(float num1, float num2)
{
    return num1 + num2;
}

/** * Function to subtract two numbers */
float sub(float num1, float num2)
{
    return num1 - num2;
}

```

```
}

/** * Function to multiply two numbers */
float mult(float num1, float num2)
{
    return num1 * num2;
}

/** * Function to divide two numbers */
float div(float num1, float num2)
{
    return num1 / num2;
}

float squa(float num1, float num2)
{
    return num1 * num2;
}
```

Sample Input and Output:

OUTPUT:

Result :

EX.NO: 5

DATE:

**CHECK IF THE GIVEN NUMBER IS ARMSTRONG NUMBER OR
NOT**

Aim :

Algorithm:

Program Logic :

Those numbers which sum of the cube of its digits is equal to that number are known as Armstrong numbers.

Example : 153

Total Digits In 153 is 3

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

$$1634 = 1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 64 = 1634$$

Examples of Armstrong numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725

Program:

```
#include<stdio.h>
int main()
{
    int num,r,sum=0,temp;
    printf("Enter a number: ");
    scanf("%d",&num);
    temp=num;
    while (num!=0)
    {
        r=num%10;
        num=num/10;
        sum=sum+(r*r*r);
    }
    if(sum==temp)
        printf("%d is an Armstrong number",temp);
    else
        printf("%d is not an Armstrong number",temp);
    return 0;
}
```

Sample input and output

Input :

Output:

Result:

EX.NO: 6

DATE:

**GIVEN A SET OF NUMBERS LIKE <10, 36, 54, 89, 12, 27>, FIND SUM
OF WEIGHTS BASED ON THE GIVEN CONDITIONS**

Aim :

Algorithm:

Program:

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
int getWeight(int n)
```

```

{
    int w=0;
    float root=sqrt(n);
    if(root==ceil(root))
        w+=5;
    if(n%4==0&&n%6==0)
        w+=4;
    if(n%2==0)
        w+=3;
    return w;
}
void main()
{
    int nums[15];
    int ws[15];
    int i,j,t,n;
    printf("Enter number of numbers : ");
    scanf("%d",&n);
    printf("\nEnter numbers : ");
    for(i=0;i<n;i++)
        scanf("%d",&nums[i]);
    for(i=0;i<n;i++)
        ws[i]=getWeight(nums[i]);
    printf("\nBefore sorting:\n");
    for(i=0;i<n;i++)
        printf("%d:%d\t",nums[i],ws[i]);
    for(i=0;i<n;i++)
        for(j=0;j<n-i-1;j++)
            if(ws[j]>ws[j+1])
            {
                t=ws[j+1];
                ws[j+1]=ws[j];
                ws[j]=t;
                t=nums[j+1];
                nums[j+1]=nums[j];
                nums[j]=t;
            }
    printf("\nSorted:\n");
}

```

```
for(i=0;i<n;i++)  
    printf("%d:%d\t",nums[i],ws[i]);  
getch();  
}
```

Sample input and output:

Input :

Output :

Result :

EX.NO 7:

DATE:

**POPULATE AN ARRAY WITH HEIGHT OF PERSONS AND FIND
HOW MANY PERSONS ARE ABOVE THE AVERAGE HEIGHT?**

Aim :

Algorithm:

Program:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 100
void readArray(int [] , int );
double findAverage(int [] , int );
int countAboveAverage(int [] , int , double);
double average;
int count;
int main(int argc, char* argv [])
{
    int arr[SIZE];
    int n;
    printf("Enter a value for n: ");
    scanf("%d", &n);
    if(n==0)
```

```

    {
        fprintf(stderr, "Your size of n is too big. Run the program
again.\n");
        exit(1);
    }
    readArray(arr, n);
    average = findAverage(arr, n);
    count = countAboveAverage(arr, n, average);
    printf("The number of persons above the average height is %d:",count);
    return EXIT_SUCCESS;
}
void readArray(int arr[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("Please enter # %d: ", i+1);
        scanf("%d", arr + i); /* or &arr[i] */
    }
}
double findAverage(int arr[], int n)
{
    int sum = 0;
    int i;
    for(i=0; i<n; i++)
        sum+=arr[i];
    return (double)sum / n;
}
int countAboveAverage(int arr[], int n, double average)
{
    int count=0;
    int i;
    for(i=0; i<n; i++)
        count = arr[i] > average ? count+1: count;
    return count;
}

```

Sample Input and output :

Input:

Output:

Result :

EX.NO 8:

DATE:

Given a string —a\$bcd./fg| find its reverse without changing the position of special characters. (Example input:a@gh%;j and output:j@hg%;a)

Aim :

Algorithm

Program :

```
#include<stdio.h>
#include<string.h>
void reverse(char str[10])
{
    int r,l;
    char c;
    r= strlen(str)-1;
    l=0;
    while(l<r)
    {
        if(!(str[l]>='a' && str[l]<='z') || (str[l]>='A' && str[l]<='Z'))
            l++;
        else if(!(str[r]>='a' && str[r]<='z') || (str[r]>='A' && str[r]<='Z'))
            r--;
    }
}
```

```
        r--;
    else
    {
        c=str[l];
        str[l]=str[r];
        str[r]=c;
        l++;
        r--;
    }
}
printf("%s",str);
}
int main()
{
//    char str1[10]="g@vat";
    char str1[10];
    printf("Enter the String to be Reverse : \n");
    scanf("%s", str1);
    printf("Reverse String is : \n");
    reverse(str1);
    return 0;
}
```

Sample Input and Output :

Result:

EX.NO :9

DATE:

**CONVERT THE GIVEN DECIMAL NUMBER INTO BINARY,
OCTAL AND HEXADECIMAL NUMBERS USING USER DEFINED
FUNCTIONS.**

Aim :

Algorithm:

Program Logic :

1. Convert the given octal number to binary number system.
2. Extract binary bits in a group of 4 bit starting from right side.
3. Write the corresponding hexadecimal of extracted 4 binary bits.

For Example

Suppose input decimal number is 13

Step 1. $13/2$, Remainder = 1, Quotient = 6

Step 2. $6/2$, Remainder = 0, Quotient = 3

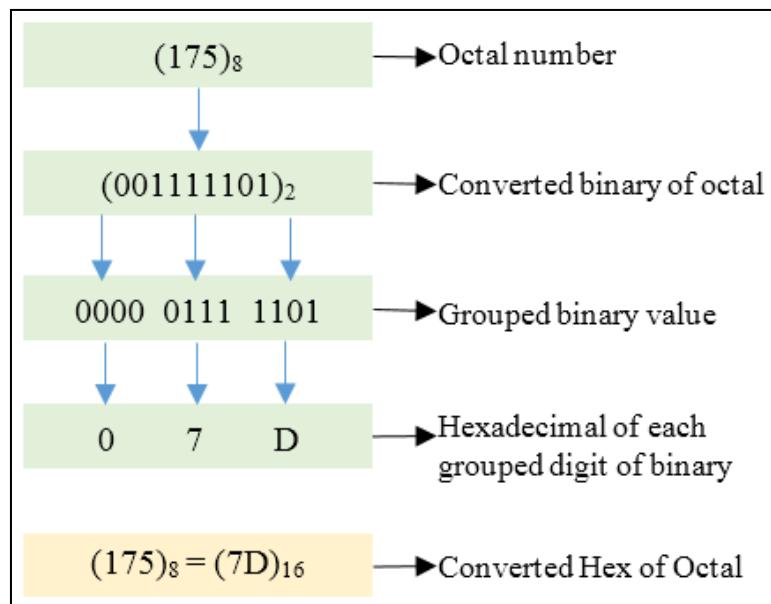
Step 3. $3/2$, Remainder = 1, Quotient = 1

Step 4. $1/2$, Remainder = 1, Quotient = 0

Now, the Binary equivalent of 13 is the remainders in reverse order :

1101

/ * Convert Octal Number To Hexadecimal Number * /



/* Decimal Number Into Binary */

Program :

/* C program to convert decimal numbers to binary numbers */

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

long decimalToBinary(long n);

void octal_hex(int n, char hex[]); int hex_octal(char hex[]);

int main()
{
    long decimal;
```

```

char hex[20],c; int n;
printf("Enter a decimal number\n");
scanf("%ld", &decimal);
printf("Binary number of %ld is %ld", decimal,
decimalToBinary(decimal));
printf("Enter octal number: ");
scanf("%d",&n);
octal_hex(n,hex);
printf("Hexadecimal number:%s",hex);
getch();
return 0;
}

/* Function to convert a decimal number to binary number */
long decimalToBinary(long n)
{
    int remainder;
    long binary = 0, i = 1;
    while(n != 0)
    {
        remainder = n%2;
        n = n/2;
        binary= binary + (remainder*i);
        i = i*10;
    }
    return binary;
}

void octal_hex(int n, char hex[])
{
    int i=0,decimal=0, rem;

```

```
while (n!=0)
{
    rem = n%10;
    n=n/10;
    decimal=decimal+rem*pow(8,i);
    ++i;
}
i=0;
while (decimal!=0)
{
    rem=decimal%16;
    switch(rem)
    {
        case 10:
            hex[i]='A';      break;
        case 11:
            hex[i]='B';      break;
        case 12:
            hex[i]='C';      break;
        case 13:
            hex[i]='D';      break;
        case 14:
            hex[i]='E';      break;
        case 15:
            hex[i]='F';      break;
        default:
            hex[i]=rem+'0';  break;
    }
    ++i;
```

```
decimal/=16;  
}  
hex[i]='\0';  
strrev(hex); /* Function to reverse string. */  
}
```

Sample Input and Output :

Input:

Output:

Result:

EX.NO.10

DATE:

**MENU PROGRAM TO PERFORM FOLLOWING OPERATION FOR
GIVEN PARAGRAPH**

Aim :

Algorithm

Program :

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char a[100];
    int len,i,word=1;
    clrscr();
    printf("\nEnter A STRING: ");
    gets(a);
```

```
len=strlen(a);
for(i=0;i<len;i++)
{
    if(a[i]!=' '&& a[i+1]==' ')
        word=word+1;
}
printf("\nTHERE ARE %d WORDS IN THE STRING",word);
getch();
}
```

Sample Input And Output:

Input:

Output:

Program to Capitalize the first word of each sentence.

Program :

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char s[100];
    int len,i;
    clrscr();
    printf("\nENTER A SENTENSE: ");
    gets(s);
    len=strlen(s);
    printf("\n");
    for(i=0;i<len;i++)
    {
        if((i==0 && s[i]>=97 && s[i]<=122) || (s[i-1]==32 && s[i]>=97 &&
s[i]<=122))
            printf("%c",s[i]-32);
        else
        {
            if(i!=0 && s[i-1]!=32 && s[i]>=65 && s[i]<=90)
                printf("%c",s[i]+32);
            else
                printf("%c",s[i]);
        }
    }
    getch();
```

}

Sample Input and Output :

Input:

Program to Replace a given word with another word

Aim:

Algorithm

Program :

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

```
void main()
```

```
{
```

```
char s[] = "All work and no play makes Jack a dull boy.";
char word[10],rpwrd[10],str[10][10];
int i=0,j=0,k=0,w,p;

printf("All work and no play makes Jack a dull boy.\n");
printf("\nEnter WHICH WORD IS TO BE REPLACED\n");
scanf("%s",word);
printf("\nEnter BY WHICH WORD THE %s IS TO BE
REPLACED\n",word);
scanf("%s",rpwrd);
p=strlen(s);

for (k=0; k<p; k++)
{
    if (s[k]!=' ')
    {
        str[i][j] = s[k];
        j++;
    }
    else
    {
        str[i][j]='\0';
        j=0; i++;
    }
}

str[i][j]='\0';
w=i;

for (i=0; i<=w; i++)
{
    if(strcmp(str[i],word)==0)
        strcpy(str[i],rpwrd);

    printf("%s ",str[i]);
}
getch();
```

}

How can I replace the word just 'Jack'? like

Sample Input and Output:

Input:

Output

Result :

EX.NO:11(A)

SORTING TECHNIQUES

DATE:

SELECTION SORT

Aim:

Algorithm:

Program :

```
#include<stdio.h>
int main()
{
int s,i,j,temp,a[20];
printf("Enter total elements: ");
scanf("%d",&s);
printf("Enter %d elements: ",s);
for(i=0;i<s;i++)
scanf("%d",&a[i]);
for(i=0;i<s;i++)
{
for(j=i+1;j<s;j++)
{
if(a[i]>a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
}
```

```
a[i]=a[j];
a[j]=temp;
}
}
}

printf("After sorting is: ");
for(i=0;i<s;i++)
printf("%d",a[i]);
return 0;
}
```

Sample Input and Output:

Result:

INSERTION SORT

Aim:

Algorithm:

Program :

```
#include <stdio.h>

void insert(int a[], int n) /* function to sort an array with insertion sort */
{
    int i, j, temp;
    for (i = 1; i < n; i++) {
        temp = a[i];
        j = i - 1;

        while(j>=0 && temp <= a[j]) /* Move the elements greater than temp to
one position ahead from their current position*/
        {
            a[j+1] = a[j];
            j = j-1;
        }
    }
}
```

```
        }
        a[j+1] = temp;
    }
}

void printArr(int a[], int n) /* function to print the array */
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}

int main()
{
    int a[] = { 12, 31, 25, 8, 32, 17 };
    int n = sizeof(a) / sizeof(a[0]);
    printf("Before sorting array elements are - \n");
    printArr(a, n);
    insert(a, n);
    printf("\nAfter sorting array elements are - \n");
    printArr(a, n);

    return 0;
}
```

Sample Input and Output:

Input:

Result:

EX.NO 11(B)

DATE:

SORT THE LIST OF NUMBERS USING PASS BY REFERENCE

Aim :

Algorithm:

Program:

```
#include<stdio.h>
#include<conio.h>
void sort(int m, int x[]);
void main()
{
    int i,n;
    int num[10];
    printf("Enter upper limit value:\n");
    scanf("%d",&n);
    printf("Enter the values to sort\n");
    for(i=0;i<n;i++)
        scanf("%d",&num[i]);
    sort(n,&num[0]);
    printf("after sorting:\n");
```

```
for(i=0;i<n;i++)
    printf("%4d",num[i]);
printf("\n");
getch();
}
void sort(int m,int x[])
{
int i,j,t;
for(i=1;i<=m-1;i++)
    for(j=1;j<=m-i;j++)
        if(x[j-1]>=x[j])
        {
            t=x[j-1];
            x[j-1]=x[j];
            x[j]=t;
        }
}
```

Sample Input and Output:

Input:

Output

Result:

EX.NO 12(A)

DATE:

PROGRAM FOR LINEAR SEARCH

Aim :

Algorithm :

Program

```
#include<stdio.h
>int main()
{
    int a[10],i, n, m, c=0;
    printf("Enter the size of an
array:");scanf("%d", &n);
```

```
printf("Enter the elements of the
array:");for(i=0;i<=n-1;i++)
{
    scanf("%d", &a[i]);
}
printf("Enter the number to be
search:");scanf("%d",&m);
for(i=0;i<=n-1;i++)
{
    if(a[i]==m)
    {
        c=1;
        break;
    }
}
if(c==0)
    printf("The number is not in the
list");else
    printf("The number is found at
%d",i);return 0;
}
```

Sample Input and Output:

Result: