

PROGRAM - I

→ WAP that calculates SI & CI

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

```
int main () {
```

```
float p, r, t, si, ci;
```

```
printf ("Enter the value of p, r, & t respectively");
```

```
scanf ("%f %f %f", &p, &r, &t);
```

```
si = p * r * t / 100;
```

```
ci = p * (1 + r / 100) ^ t - p;
```

```
printf ("Simple interest = %d ", si);
```

```
printf ("Compound interest = %d ", ci);
```

```
return 0;
```

```
}
```

Output :-

Enter the value of p, r, t respectively = 2, 3, 5

$$SI = 0 \cdot 3$$

$$CI = 0 \cdot 3185$$

PROGRAM-2

→ WAP that accepts the temp in $^{\circ}\text{C}$ & convert into F using the formula $\text{C}/5 = (\text{F}-32)$

```
#include<stdio.h>
int main () {
    float f, c;
    printf (" Enter the temp in centigrade ");
    scanf ("%f", &f);
    f = ((c * 9) / 5) + 32;
    printf (" temp in fahrenheit = %f ", f);
    return 0;
}
```

Output :-

Enter the temp in centigrade - 40
Temp in fahrenheit = 104.00000

PROGRAM-3

→ WAP that swaps value of two variables using a third variable

```
#include<stdio.h>
int main ()
{
    int a,b,c;
    printf ("Enter the first no = ");
    scanf ("%d", &a);
    printf ("Enter the second no. = ");
    scanf ("%d", &b);
    c = a;
    a = b;
    b = c;
    printf ("first number = %d", a);
    printf ("second number = %d", b);
    return 0;
}
```

Programs

extra feature of C is that it provides
conversion of one type of data to another type

Output :-

Enter the first no. = 10

Enter the second no. = 20

first number = 20

second number = 10

PROGRAM - 7

→ WAP to find the greatest of three no' using condition operation f nested if else

Conditional operation

```
#include <stdio.h>
int main () {
    int a,b,c;
    printf ("enter three no. ");
    scanf ("%d %d %d", &a, &b, &c);
    greatest = a > b ? (a > c ? a : c) : (b > c ? b : c);
```

```
    printf ("the greatest no. is %d", greatest);
    return 0;
}
```

(iii) by nested if else if :-

```
#include <stdio.h>
int main()
{
    int a,b,c;
    printf("Enter three values");
    scanf("%d%d%d", &a, &b, &c);
    if (a>b && a>c)
        printf("%d is greatest", a);
    else if (b>c)
        printf("%d is greatest", b);
    else
        printf("%d is greatest", c);
    return 0;
}
```

PROGRAM

Output:-

Enter three no. - 4, 5, 2
The greatest no is 5

PROGRAMS

→ WAP that accept mark of five subject & find percentage and print grades according to the right case

```
#include <stdio.h>
int main ()
{
    int maths, che, phy, bio, hindi, percent;
    printf ("Enter the marks of maths, che, phy, bio, hindi");
    scanf ("%d%d%d%d%d", &maths, &che, &phy, &bio, &hindi);
    percent = (maths + phy + che + bio + hindi) / 5
    if (percent <= 100 && percent >= 90)
        printf ("Grade A");
    else if (percent >= 80)
        printf ("Grade B");
    else if (percent < 80 && percent >= 60)
        printf ("Grade C");
    else
        printf ("Your grade is D");
    return 0;
}
```

Output :-

Enter marks of maths ,che, phy, bio ,hindi =

90 30 80 100 - 100

Percent = 80

Grade B

PROGRAM-6

→ WAP that takes two operands & one operator from the user and perform the operation using switch statement.

```
#include <stdio.h>
int main ()
{
    char ch;
    int a,b,c;
    printf ("Enter the value of a & b:");
    scanf ("%d %d", &a, &b);
    printf ("Enter your operations : (+,-,*,\")");
    scanf ("%c", &ch);
    switch (ch)
```

Case '+': c = a+b;

```
    printf (" the sum is %d", c);
    break;
```

Case '-': c = a-b;

```
    printf (" subtraction is %d", c);
    break;
```

Case '*': c = a*b;

```
    printf (" product is %d", c);
    break;
```

```
case '1': c = a/b;
printf ("division is %d", c);
break;

default : printf ("Error! plz try later 'J';
break;

return 0;
}
```

2-07-2029

total of marks out of 2000 is 976
you have got 976 marks in total

Output :-

Enter the value of a $f b = 4 \ 6$

Enter your operators : *
the product is 24

* WAP to check whether the entered number is prime or not.

```
#include <stdio.h>
int main() {
    int num, flag = 0;
    printf("Enter any number");
    scanf("%d", &num);
    for (int i = 2; i < num/2; i++) {
        if (num % i == 0)
            flag = 1;
        break;
    }
    if (flag == 0)
        printf("%d is a prime number", num);
}
```

Output

Enter any number 41
41 is a prime number

* WAP to find Reverse of a number

```
#include <stdio.h>
int main() {
    int n, reverse = 0, remainder;
    printf("Enter an integer: ");
    scanf("%d", &n);
    while (n != 0) {
        remainder = n % 10;
        reverse = reverse * 10 + remainder;
        n = n / 10;
    }
    printf("Reversed number = %d", reverse);
    return 0;
}
```

Output

Enter an integer: 8654
Reversed number = 4568

q) WAP to print Armstrong number from 1 to 100.

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

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

```
int main()
```

```
{ int a, b, c;
```

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

```
int p = 0, ans[p];
```

```
printf ("All Armstrong number between 1 and %d are : \n", a)
```

```
for (int i = 1; i < a; i++)
```

```
{
```

```
int an = 0, c = i, b = i, n = 0;
```

```
while (b > 0)
```

```
{
```

```
    n++;
```

```
    b = b / 10;
```

```
}
```

```
    while (c > 0)
```

```
{
```

```
        an = an + (pow (c / 10, n));
```

```
        c = c / 10;
```

```
}
```

```
    if (an == i)
```

```
{
```

```
        ans[p] = i;
```

Date _____

Expt. No. _____

Page No. _____

p = p + 1;

}

}

for (int i = 0; i < p; i++)

{

printf ("%d", ans[i]);

}

Page Signature : _____

and output Armstrong numbers between 1 and 100.

Output

100

All Armstrong numbers between 1 and 100 are:

1 2 3 4 5 6 7 8 9

Date _____

Expt. No. 10

Page No. _____

10 WAP to print the fibonacci series using recursion

```
#include <stdio.h>
int fib(int);
void main()
{
    int n, i, f;
    printf ("Enter the no. of terms");
    scanf ("%d", &n);
    for(i=0; i<n; i++)
    {
        f = fib(i);
        printf ("%d\n", f);
    }
}
```

```
int fib(int n)
{
    int f;
    if (n==0 || n==1)
        return n;
    else
        f = fib(n-1) + fib(n-2);
    return f;
}
```

Teacher's Signature : _____

Output

```
Enter the no. of terms 10  
0 1 1 2 3 5 8 13 21 34
```

Program 11.

WAP to search an element in an array using Linear Search.

Source Code:

```
#include <stdio.h>
void main()
{
    int a[10];
    int i, n, item, found = 0;
    printf("Enter the number of elements in array: \n");
    scanf("%d", &n);
    printf("Enter the elements one by one:");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Enter the element to be searched: ");
    scanf("%d", &item);
    /* Linear search begins */
    for (i = 0; i < n; i++)
    {
        if (item == a [i] )
        {
            found = 1;
            printf("Element is present in the array at position = %d\n", i+1);
            break;
        }
    }
    if (found == 0)
        printf("Element is not present in the array\n");
}
```

OUTPUT:

Enter the number of elements in array:

8

Enter the elements one by one: 23 34 45 2 12 1 87 50

Enter the element to be searched: 12
Element is present in the array at position = 5

Program 12.

WAP to sort the elements of the array in ascending order using

(a)Bubble Sort

(b)Selection Sort

(a) Source Code for Bubble sort:

```
#include <stdio.h>
int main()
{
    int array[100], n, i, j, swap;
    printf("Enter number of elements\n");
    scanf("%d", &n);
    printf("Enter array elements\n");
    for (i = 0; i < n; i++)
        scanf("%d", &array[i]);
    /*BUBBLE SORT BEGINS*/
    for (i = 0 ; i < ( n - 1 ); i++)
    {
        for (j = 0 ; j < n-i-1; j++)
        {
            if (array[j] > array[j+1])
            {
                swap = array[j];
                array[j] = array[j+1];
                array[j+1] = swap;
            }
        }
    }
    printf("Sorted list in ascending order:\n");
    for (i = 0 ; i < n ; i++)
        printf("%d\t", array[i]);
```

```
    return 0;  
}
```

OUTPUT:

```
Enter number of elements  
8  
Enter array elements  
23 3 99 45 12 9 34 5  
Sorted list in ascending order:  
3      5      9      12      23      34      45      99
```

(b) Source Code for Selection sort:

```
#include <stdio.h>  
  
int main()  
{  
    int a[100], n, i, j, swap;  
  
    printf("Enter number of elements in array: ");  
    scanf("%d", &n);  
  
    printf("Enter array elements:\n");  
  
    for (i = 0; i < n; i++)  
        scanf("%d", &a[i]);  
  
    //SELECTION SORT BEGINS  
  
    for(i = 0; i < n - 1; i++)  
    {  
        for(j = i + 1; j < n; j++)  
        {  
            if(a[i] > a[j])  
            {  
                swap = a[i];  
                a[i] = a[j];  
                a[j] = swap;  
            }  
        }  
    }  
  
    printf("Sorted Array:\n");
```

```

for(i = 0; i < n; i++)
    printf("%d \t", a[i]);
return 0;
}

```

OUTPUT:

```

Enter number of elements in array: 8
Enter array elements:
13 4 35 11 9 56 23 5
Sorted Array:
4      5      9      11      13      23      35      56
Program 13.

```

WAP to swap 2 elements using the concept of pointers.

Source Code:

```

#include <stdio.h>
void main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\nx = %d\ny = %d\n", x, y);

    a = &x;      //assigning address of x into pointer a
    b = &y;      //assigning address of y into pointer b

    temp = *b;
    *b = *a;
    *a = temp;

    printf("After Swapping\nx = %d\ny = %d\n", x, y);
}

```

OUTPUT:

```

Enter the value of x and y
44 99
Before Swapping
x = 44
y = 99
After Swapping

```

```
x = 99
y = 44
```

Program 14.

WAP to copy contents of one file into another file. Also count no. of characters, words and lines in the file.

Source Code:

```
#include <stdio.h>
#include <stdlib.h>
void main( )
{
    FILE *fs, *ft ;
    char ch ;
    int characters = 0, words = 0, lines = 0;
    fs = fopen ("pr1.c", "r") ;
    if ( fs == NULL )
    {
        printf ( "Cannot open source file" ) ;
        exit(1);
    }
    ft = fopen ( "pr2.c", "w" ) ;
    if ( ft == NULL )
    {
        printf ( "Cannot open target file" ) ;
        fclose ( fs ) ;
        exit(1);
    }
    while ( 1 )
    {
        ch = fgetc ( fs ) ;
        if ( ch == EOF )
            break ;
        else
            fputc ( ch, ft ) ;
            characters++;
        /* Check new line */
        if (ch == '\n' || ch == '\0')
            lines++;
        /* Check words */
        if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
            words++;
    }
}
```

```
printf("Files have been copied successfully");
printf("Total characters = %d\n", characters);
printf("Total words = %d\n", words);
printf("Total lines = %d\n", lines);
fclose ( fs );
fclose ( ft );
}
```

OUTPUT:

Files have been copied successfully

Total characters = 106

Total words = 18

Total lines = 3

Program 15.

WAP to compare the contents of two files and determine whether they are same or not.

Source Code:

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

```
int main() {
    FILE *fp1, *fp2;
    int ch1, ch2;

    fp1 = fopen("PR1.c", "r");
    fp2 = fopen("PR2.c", "r");

    if (fp1 == NULL) {
        printf("Cannot open first file for reading");
        exit(1);
    }

    else if (fp2 == NULL) {
        printf("Cannot open second file for reading");
        exit(1);
    }

    else {
        ch1 = getc(fp1);
        ch2 = getc(fp2);
```

```
while ((ch1 != EOF) && (ch2 != EOF) && (ch1 == ch2)) {  
    ch1 = getc(fp1);  
    ch2 = getc(fp2);  
}  
  
if (ch1 == ch2)  
    printf("Files are identical ");  
else if (ch1 != ch2)  
    printf("Files are Not identical ");  
  
fclose(fp1);  
fclose(fp2);  
}  
return (0);  
}
```

OUTPUT:

Files are identical