

② WAP to find transpose of a matrix.

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

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

```
void main ()
```

```
{ int a[2][2], i, j;
```

```
printf("display the enter matrix\n");
```

```
printf("enter the element of matrix\n");
```

```
for (i=0; i<2; i++)
```

```
{
```

```
for (j=0; j<2; j++)
```

```
{ scanf("%d", &a[i][j]);
```

```
}
```

```
}
```

```
printf("enter the matrix\n");
```

```
for (i=0; i<2; i++)
```

```
{
```

```
for (j=0; j<2; j++)
```

```
{ printf("%d\n", a[i][j]);
```

```
}
```

```
}
```

```
printf("Transpose of given matrix\n");
```

```
for (i=0; i<2; i++)
```

```
{
```

```
for (j=0; j<2; j++)
```

```
{
```

```
printf("%d\n", a[j][i]);
```

```
}
```

```
}
```

getch ();

3

Output :-

enter the matrix

$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

Transpose of given matrix

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

③ WAP to find the addition of two matrix.

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

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

```
void main ()
```

```
{
```

```
    int m, n, i, j, a[2][2], b[2][2];
```

```
    printf("enter the no. of row and column\n");
```

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

```
    printf("enter the element of matrix a\n");
```

```
    for (i=0; i<m; i++)
```

```
    {
        for (j=0; j<n; j++)
```

```
        {
            scanf("%d", &a[i][j])
```

```
        }
```

```
    }
```

```
    printf("enter the element of matrix b\n");
```

```
    for (i=0; i<m; i++)
```

```
    {
        for (j=0; j<n; j++)
```

```
        {
            scanf("%d", &b[i][j]);
```

```
        }
```

```
    }
```

```
    printf("sum of entered matrix\n");
```

```
    for (i=0; i<m; i++)
```

```
    {
        for (j=0; j<n; j++)
```

```
        {
            printf("%d\t", a[i][j]+b[i][j]);
```

```
5 printf("don");
```

```
3 getch();
```

```
3
```

Output :- Enter the no. of row and column

2 → Row

2 → Column

Entered Element of matrix a

$$\begin{bmatrix} 2 & 4 \\ 3 & 5 \end{bmatrix}$$

Entered element of matrix b

$$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix}$$

Sum of entered matrix

$$\begin{bmatrix} 2 & 6 \\ 4 & 8 \end{bmatrix}$$

④ WAP in C of linear search an element in given list.

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

int main ()
{
    int a[10], i, item, n;
    printf("Enter the size of an array\n");
    scanf("%d", &n);
    printf("Enter element of array\n");
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);
    printf("Enter item to search\n");
    scanf("%d", &item);
    for (i=0; i<=10; i++)
        if (item == a[i])
        {
            printf("item found at location %d\n", i+1);
            break;
        }
    if (i == n)
    {
        printf("item does not exist\n");
        return 0;
    }
}
```

Output :-

Enter size of an array: 4

Enter element:

4

6

7

2

Enter item to search

2

Item found at location 4.

⑤ WAP in C of Binary Search. an element in a given list.

```
#include <stdio.h>
#include <conio.h>
int main ( )
{
    int a[100], i, item, n, low, high, mid;
    printf("Enter the size of an array\n");
    scanf("%d", &n);
    printf("Enter element of array:\n");
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);
    printf("Enter item to search\n");
    scanf("%d", &item);
    low = 0;
    high = n-1;
    mid = (low+high)/2;
    while (low <= high)
    {
        if (a[mid] < item)
        {
            low = mid+1;
        }
        else if (a[mid] == item)
        {
            printf("%d found at location %d.\n", item, mid+1);
            break;
        }
    }
```

else

{

high = mid - 1 ;

mid = (low + high) / 2 ;

}

if (low > high)

{

printf("Not found! %d is not present in list.\n", item);

}

return 0 ;

}

Output :-

Enter the size of an array : 4

enter element of an array:

3

2

1

0

enter item to search

2

item found at location. 2

⑥ WAP in C to traverse an element of an array.

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

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

```
int main()
```

```
{
```

```
int a[10], i, n;
```

```
printf("Enter the size of an array: \n");
```

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

```
printf("Enter the element \n");
```

```
for(i=0; i<n; i++)
```

```
{
```

```
scanf("%d", &a[i]);
```

```
}
```

```
printf("traverse element \n");
```

```
for(i=0; i<n; i++)
```

```
{
```

```
printf("%d", a[i]);
```

```
}
```

```
return 0;
```

```
}
```

Output :-

Enter size of an array :

5

enter the element

3

4

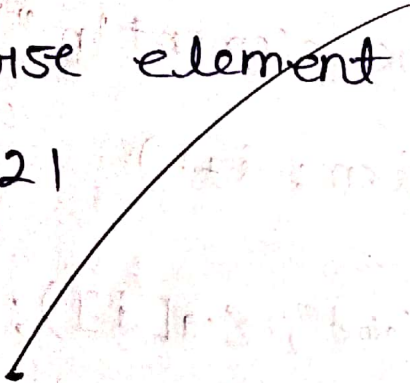
5

2

1

~~traverse element~~

34521



⑦ WAP in C to find multiplication of two matrices

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

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

```
void main ()
```

```
{
```

```
int m, n, i, j, a[2][2], b[2][2];
```

```
printf("enter the no. of row and column\n");
```

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

```
printf("enter the element of matrix a\n");
```

```
for (i=0; i<m; i++)
```

```
{  
for (j=0; j<n; j++)
```

```
{  
scanf("%d", &a[i][j])
```

```
}
```

```
}
```

```
printf("enter the element of matrix b\n");
```

```
for (i=0; i<m; i++)
```

```
{  
for (j=0; j<n; j++)
```

```
{  
scanf("%d", &b[i][j]);
```

```
}
```

```
}
```

```
printf("multiply of entered matrix\n");
```

```
for (i=0; i<m; i++)
```

```
{  
for (j=0; j<n; j++)
```

```
{  
printf("%d\t", a[i][j] * b[i][j]);
```

```
3 printf("%d\n");
```

```
3 getch();
```

```
3
```

Output :-

enter the no. of row and column

2

2

enter the element of matrix a:

2

1

3

4

enter the element of matrix b:

0

2

3

1

Multiply of the entered matrix

0

2

9

4

Program-08

NAP in C to insert the item in stack-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[50], i, n, maxsize, item, top;
    printf("Enter the no. of elements\n");
    scanf("%d", &n);
    printf("Enter the elements of an array\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    printf("Element of an array\n");
    for(i=0; i<n; i++)
        printf("%d\n", a[i]);
    printf("Enter the element to insert\n");
    scanf("%d", &item);
    maxsize = 50;
    top = n-1;
    if (top == maxsize-1)
    {
        printf("Stack is full");
    }
    else
    {
        top = top+1;
        a[top] = item;
        n = n+1;
        for(i=0; i<n; i++)
            printf("%d", a[i]);
    }
    return 0;
}
```

Output:-

Enter the no. of elements

4

Enter the elements of an array

7

9

4

6

Element of an array

7

9

4

6

Enter the element to insert

8

7 9 4 6 8

Program-09

WAP in C to delete the item in stack-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[50], i, n, top, item, maxsize;
    printf("Enter the no. of elements |n");
    scanf("%d", &n);
    printf("Enter the elements of an array |n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    printf("Elements of an array |n");
    for(i=0; i<n; i++)
        printf("%d |n", a[i]);
    top = n-1;
    maxsize = n;
    if(top == -1)
    {
        printf("Stack is initially empty |n");
    }
    else
    {
        item = a[top];
        top = top-1;
        n = n-1;
        printf("Last element will delete and the array is: |n");
        for(i=0; i<n; i++)
            printf("%d ", a[i]);
    }
    return 0;
}
```

Output =

Enter the no. of elements

3
Enter the elements of an array

6

7

4

Elements of an array

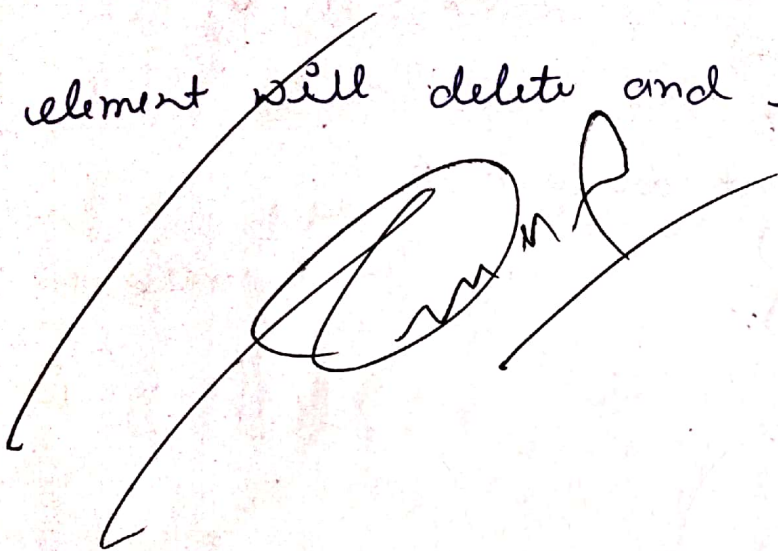
6

7

4

Last element will delete and the array is:

67



Program:-10

WAP In C to find factorial of any number.

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

```
int fact (int n);
```

```
void main ()
```

```
{  
    int n, factorial;
```

```
    printf("find the factorial of any number\n");
```

```
    printf("enter the number\n");
```

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

```
    f = fact(n);
```

```
    printf("factorial = %d", factorial);
```

```
    getch();
```

```
}
```

```
int fact (int n)
```

```
{
```

```
    if (n == 0)
```

```
        return 1;
```

```
    else
```

```
        return (n * fact(n-1));
```

```
}
```

output :-

Enter the number

0

factorial = 1

Enter the number

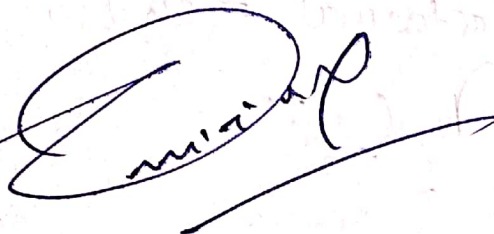
1

factorial = 1

Enter the number

3

factorial = 6



Program 8-11

WAP in C to find the value of Ackermann function

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

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

```
int Ack (int m, int n);
```

```
void main ()
```

```
{  
    int m, n;
```

```
    printf ("find the value of Ackermann function\n");
```

```
    printf ("enter the value of m: and n: \n");
```

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

```
    printf ("Ackermann value = %d\n", Ack(m, n));
```

```
    getch ();
```

```
}
```

```
int Ack (int m, int n)
```

```
{
```

```
    if (m == 0)
```

```
        return (n+1);
```

```
    else if (m > 0 && n == 0)
```

```
        return (Ack(m-1, 1));
```

```
    else if (m > 0 && n > 0)
```

```
        return (Ack(m-1, A(m, n-1)));
```

```
}
```

Output :-

Enter the value of m: and n:

ACK(0,1)

Ackermann value = 2

Enter the value of m: and n:

ACK(0,2)

Ackermann value = 3

minip

Program :- 12

WAP in C to Tower of Hanoi problem.

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

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

```
void TOH(int n, char A, char B, char C);
```

```
int main ()
```

```
{  
    int n;
```

```
    printf("enter the number of discs: \n");
```

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

```
    TOH (n, 'A', 'B', 'C');
```

```
    return 0;
```

```
}
```

```
void TOH (int n, char A, char B, char C)
```

```
{  
    if (n > 0)
```

```
{  
        TOH (n-1, A, C, B)
```

```
        printf(" \n %C → %C", A, B);
```

```
        TOH (n-1, C, B, A)
```

```
    }
```

```
}
```

Output :-

Enter the number of discs:

$n = 3$

$A \rightarrow C$

$A \rightarrow B$

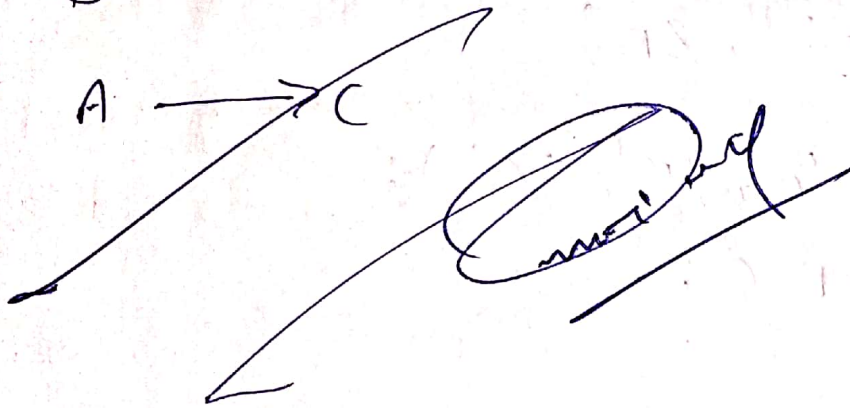
$C \rightarrow B$

$A \rightarrow C$

$B \rightarrow A$

$B \rightarrow C$

$A \rightarrow C$



Program:-13

WAP in C to insert or rear a element in Queue.

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

int main ()
{
    int Q[50], i, n, rear, item, maxsize;
    printf("Enter the size of an array\n");
    scanf("%d", &n);
    printf("Enter the elements of an array\n");
    for (i=0; i<n; i++)
        scanf("%d", &Q[i]);
    printf("Element of an array\n");
    for (i=0; i<n; i++)
        printf("%d", Q[i]);
    rear = n-1;
    maxsize = 50;
    printf("Enter the element to insert\n");
    scanf("%d", &item);
    if (rear == maxsize-1)
    {
        printf("Stack is already full\n");
    }
    else
```

```

2
year = year + 1;
a[year] = item;
n = n + 1;
for (i = 0; i < n; i++)
    printf("%d", a[i]);
}
return 0;
}

```

Output :-

Enter size of an array

4

Enter the element of an array

4

3

2

1

elements of an array

4

3

2

1

Enter item to insert.

5

43215

Program :- 14

WAP in to delete or front element from Queue.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int Q[50], i, n, front, item, maxsize;
    printf("enter the size of an array\n");
    scanf("%d", &n);
    printf("enter the elements of an array\n");
    for (i=0; i<n; i++)
        scanf("%d", &Q[i]);
    printf("enter the element of array\n");
    for (i=0; i<n; i++)
        printf("%d", Q[i]);
    front = n-1;
    maxsize = n;
    printf("enter the element to delete\n");
    scanf("%d", &item);
    if (front == -1)
    {
        printf("stack is initially empty\n");
    }
    else
```

```

}
item = Q[front];

front = front - 1;
n = n - 1;
for (i = 0; i < n; i++)
    printf("%d", Q[i]);
}
return 0;
}

```

Output :-

Enter the size of an array

3

Enter the element of an array

2
0
1

Element of an array

2
0
1

Enter item to delete → 1

20

minip

Program:- 15

WAP in C of Bubble sorting

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

int main ()
{
    int a[10], i, n, j, temp;

    printf("enter the size of an array: \n");
    scanf("%d", &n);

    printf("enter the element \n");

    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }

    printf("%d \n sorted in ascending order");
}
```

```

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

```

Output :-

Enter the size of an array:

4

enter the element of array

0

8

9

7

Sorted in ascending order 0789

Program :- 16

WAP in C of selection sorting.

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    int a[10], i, n, j, temp;
    printf("enter the size of an array\n");
    scanf("%d", &n);
    printf("enter the element\n");
    for (i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    for (i=0; i<n; i++) // min=i;
    {
        for (j=i+1; j<n; j++) ✓
        {
            if (a[j] < a[i]) // min=j;
            {
                if (min != i)
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
    }
}
```

```
printf("%i\n sorted in ascending order");
```

```
for (i=0; i<n; i++)
```

```
{  
    printf("%i", a[i]);
```

```
}
```

```
return 0;
```

```
}
```

output :-

Enter size of an array :

4

enter the element

0

9

7

8

sorted in ascending order 0789

Pranav

Program :- 17

WAP in C of Insertion sort.

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

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

```
int main()
```

```
{
```

```
    int a[50], i, j, n, temp;
```

```
    printf("Enter the number of elements\n");
```

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

```
    printf("Enter the elements\n");
```

```
    for (i=0; i<n; i++)
```

```
    {
```

```
        scanf("%d", &a[i]);
```

```
    }
```

```
    for (i=1; i<n; i++)
```

```
    {
```

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

```
        j = i-1;
```

```
        while ((j >= 0) && (a[j] > temp))
```

```
        {
```

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

```
            j = j-1;
```

```
        }
```

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

```
    }
```

```
    printf("Display of sorted order\n");
```

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

```
        printf("%d", a[i]);
```

```
return 0;  
}
```

Output :-

Enter the size of array:-
4

Enter the element

5

9

0

2

Display sorted array

0259



WAP of Quick sorting

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

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

```
void quicksort (int a[100], int low, int high);
```

```
int main ( )
```

```
{
```

```
    int i, n, a[100];
```

```
    printf (" enter the size of array \n ");
```

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

```
    printf (" enter the elements of array: \n ");
```

```
    for (i=0; i<n; i++)
```

```
        scanf ("%d", &a[i]);
```

```
    quicksort (a, 0, n-1);
```

```
    printf ("sorted in ascending order :-> ");
```

```
    for (i = 0 ; i < n ; i++)
```

```
        printf ("%d ", a[i]);
```

```
    return 0;
```

```
}
```

```
void quicksort (int a[100], int low, int high)
```

```
{
```

```
    int i, j, pivot, temp;
```

```
    if (low < high)
```

```
    {
```

```
        pivot = low;
```

```
        i = low;
```

```
        j = high;
```

```

while (i < j)
{
    while (a[i] <= a[p] && i < high)
        i++;
    while (a[j] > a[p])
        j--;

```

```

    if (i < j) // swap a[i] ↔ a[j]

```

```

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

```

```

}

```

```

} else // swap a[p] ↔ a[j]

```

```

{
    temp = a[p];
    a[p] = a[j];
    a[j] = temp;
}

```

```

    QuickSort(a, low, j-1);

```

```

    QuickSort(a, j+1, high);

```

```

}

```

Output

Enter the size of array:→

4

Enter the element: 8 5 4 2

Order of sorted elements:-

2 4 5 8

A stylized handwritten signature in black ink, consisting of a large loop followed by a horizontal stroke and a final upward stroke.

Program :-19

WAP in C for Mergesort.

```
#include <stdio.h>
#include <conio.h>
void mergesort(int a[], int low, int high);
void combine (int a[], int low, int mid, int high);
void main ()
{
    int a[20], i, n, low, high;
    printf("enter the size of array:\n");
    scanf("%d", &n);
    printf("enter the element of array:\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    getch();
}

void mergesort(int a[], int low, int high)
{
    int mid;
    if (low < high)
    {
        mid = (low+high)/2;
        mergesort(a, low, mid);
        mergesort(a, mid+1, high);
        combine(a, low, mid, high);
    }
}

void combine (int a[], int low, int mid, int high).
```

```

{
    int temp[50];
    int i, j, k;
    i = low;
    j = mid+1;
    k = low;
    while (i ≤ mid && j ≤ high)
    {
        if (a[i] ≤ a[j])
            temp[k++] = a[i++];
        else
            temp[k++] = a[j++];
    }
if * while (i ≤ mid)
    {
        temp[k++] = a[i++];
    }

```

```

{
    while (j ≤ high)
        temp[k++] = a[j++];
}
for (i = low; i ≤ high; i++)
    a[i] = temp[i];
}

```

Output :-

Enter the size of array: 4

Enter the element of array:
4
9
0
4

Sorted array is → 0449

Program :- 20

WAP in C to create a linked list and display it.

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

```
#include <stdlib.h>
```

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node * next;
```

```
};
```

```
int main()
```

```
struct node * node1, * node2, * node3;
```

```
// Create space for node
```

```
node1 = (struct node *) malloc (sizeof (struct node));
```

```
node2 = (struct node *) malloc (sizeof (struct node));
```

```
node3 = (struct node *) malloc (sizeof (struct node));
```

```
printf("enter the data of linked list = ");
```

```
scanf("%d %d %d", &node1->data, &node2->data, &node3->data);
```

```
// Link the node
```

```
node1->next = node2;
```

```
node2->next = node3;
```

```
node3->next = 0;
```

```
// To display
```

```
struct node * head;
```

```
head = node1;
```

```
while (head->next != 0)
```

```
{ printf("%d -> ", head->data);
```

```
    head = head->next;
```

```
printf(" %d", head → data);  
return 0;
```

Output :-

Enter the data of linked list =
8 9 6

8 → 9 → 6 → ~~NULL~~

