

```

#include <iostream>

using namespace std;

const int size1 = 8;
const int size2 = 5;

void con_stack(int stack[size2], int &top, int q[size1], int &f, int &r)
{
    int i, value;

    for(i = 0; i <= r; i++)
    {
        if(q[i] > 100)
        {
            if(top == size1 - 1)
            {
                value = stack[top];
                --top;

                ++top;
                stack[top] = q[i];
            }
            else
            {
                ++top;
                stack[top] = q[i];
            }
        }
    }
}

void prints(int stack[size2], int top)
{
    int i;
    cout<<"stack larger 100 : \n";
    for(i = 0; i <= top; i++)
        cout<<stack[i]<<"\t";
    cout<<"\n";
}

void printq(int q[size1], int f, int r)
{
    int i;
    cout<<"Queue elements : \n";

    if(f == -1)
        cout<<"Queue is empty nothing to print !!";

    else
    {
        for(i = r; i >= f; i--)
        {
            cout<<q[i]<<"\t";
        }
    }
}

```

```
int main()
{
    int q[size1]={100, 203, 99, 409, 523, 611, 79};
    int stack[size2];
    int top = -1;
    int f = 0;
    int r = 6;

    printq(q, f, r);
    cout<<"\n";

    con_stack(stack, top, q, f, r);
    cout<<"\n";

    prints(stack, top);
    cout<<"\n";

    return 0;
}
```

```

#include <iostream>

using namespace std;

const int size = 6;

void insertion(int q[size], int &f, int &r)
{
    int value;
    if(r == size - 1)
        cout<<"Queue is full !! Insertion is not possible.\n";

    else
    {
        cout<<"Enter any number : ";
        cin>>value;

        ++r;
        q[r] = value;
    }

    if(f == -1)
        f = 0;
}

void deletion(int q[size], int &f, int &r)
{
    int value;
    if(f == -1)
        cout<<"Under flow !! Queue is empty deletion is not possible.\n";

    else if(f == r)
    {
        value = q[f];

        f = -1;
        r = -1;
    }

    else
    {
        value = q[f];
        ++f;
    }
}

void print(int q[size], int f, int r)
{
    int i;

    if(f == -1)
        cout<<"Queue is empty nothing to print !!";

    else
    {
        for(i = r; i >= f; i--)
        {
            cout<<q[i]<<"\t";
        }
    }
}

```

```
    }  
}
```

```
int main()  
{  
    int q[size];  
    int f = -1;  
    int r = -1;  
    int x;  
  
    for( ; x != 4; )  
    {  
        cout<<"1- insertion\n";  
        cout<<"2- deletion\n";  
        cout<<"3- print\n";  
        cout<<"4- exit\n";  
  
        cout<<"Enter your choice : ";  
        cin>>x;  
        cout<<"\n";  
  
        switch (x)  
        {  
            case 1 :insertion(q, f, r); break;  
            case 2 :deletion(q, f, r); break;  
            case 3 :print(q, f, r); break;  
            default:cout<<"Error\n";  
        }  
        cout<<"\n";  
    }  
  
    return 0;  
}
```

Q1/ Having a Queue of size (10), write complete program to split the even values of the queue into other queue of size (5) and the odd values into stack of size (5).

```
-----

#include <iostream>

using namespace std;

int const sizeq1 = 10;
int const sizeq2 = 5;
int const sizest = 5;

void insertion(int q1[sizeq1], int &f1, int &r1)
{
    int value;
    int i;

    for(i = 0; i <= 9; i++)
    {
        if(r1 == sizeq1-1)
            cout<<"Queue is empty, insertion is not possible\n";

        else
        {
            cout<<"Enter any number for queue : ";
            cin>>value;

            ++r1;
            q1[r1] = value;
        }

        if(f1 == -1)
            f1 = 0;
    }
}

void spillt_qu_st(int q1[sizeq1], int q2[sizeq2], int stack[sizest], int &top, int
&f1, int &r1, int &f2, int &r2)
{
    int i;

    for(i = r1; i >= f1; i--)
    {
        if(q1[i] % 2 == 0)
        {
            r2++;
            q2[r2] = q1[i];

            if(f2 == -1)
                f2 = 0;
        }

        else if(q1[i] % 2 != 0)
        {
            top++;
            stack[top] = q1[i];
        }
    }
}
```

```

    }
}

void printq1(int q1[sizeq1], int f1, int r1)
{
    int value;
    int i;

    cout<<"\n Queue 1 full : \n";

    if(r1 == -1)
        cout<<"Queue 1 is empty\n";
    else
    {
        for(i = r1; i >= f1; i--)
            cout<<q1[i]<<"\t";
    }
}

void printq2(int q2[sizeq2], int f2, int r2)
{
    int value;
    int i;

    cout<<"\n Queue 2 even : \n";

    if(r2 == -1)
        cout<<"Queue 2 is empty\n";
    else
    {
        for(i = r2; i >= f2; i--)
            cout<<q2[i]<<"\t";
    }
}

void printst(int stack[sizest], int top)
{
    int value;
    int i;

    cout<<"\n stack odd : \n";
    if(top == -1)
        cout<<"stack 1 is empty\n";
    else
    {
        for(i = 0; i <= top; i++)
            cout<<stack[i]<<"\t";
    }
}

int main()
{
    int q1[sizeq1];
    int q2[sizeq2];
    int stack[sizest];

```

```
int top = -1;

int f1 = -1;
int r1 = -1;

int f2 = -1;
int r2 = -1;

insertion(q1, f1, r1);
cout<<"\n";

spillt_qu_st(q1, q2, stack, top, f1, r1, f2, r2);
cout<<"\n";

printq1(q1, f1, r1);
cout<<"\n";

printq2(q2, f2, r2);
cout<<"\n";

printst(stack, top);
cout<<"\n";

return 0;
}
```

Q5/ Given Queue of size (5) with (5) elements, find the factorial of each value of this queue and put it in an empty stack of size (5) ?

```
-----

#include <iostream>

using namespace std;

int const size = 5;

void insertion(int q[size], int &f, int &r)
{
    int value;
    int i;

    for(i = 0; i <= 4; i++)
    {
        if(r == size-1)
            cout<<"Queue is empty, insertion is not possible\n";

        else
        {
            cout<<"Enter any number for queue : ";
            cin>>value;

            ++r;
            q[r] = value;
        }

        if(f == -1)
            f = 0;
    }
}

void print(int q[size], int f, int r)
{
    int value;
    int i;

    if(r == -1)
        cout<<"Queue is empty\n";
    else
    {
        for(i = r; i >= f; i--)
            cout<<q[i]<<"\t";
    }
}

void qu_st_fc(int q[size], int stack[size], int &top, int &f, int &r)
{
    int i, j, x;
    int fact = 1;

    for(i = r; i >= f; i--)
    {
        fact = 1;
```



```

        x = q[i];
        for(j = 1; j <= x; j++)
        {
            fact = fact * j;
        }

        ++top;
        stack[top] = fact;
    }
}

void printst(int stack[size], int top)
{
    int i;

    if(top == -1)
        cout<<"Stack is empty !!\n";
    else
    {
        for(i = 0; i <= top; i++)
            cout<<stack[i]<<"\t";
    }
}

int main()
{
    int q[size];
    int stack[size];

    int top = -1;
    int f = -1;
    int r = -1;

    insertion(q, f, r);
    cout<<"\n Queue\n";

    print(q, f, r);
    cout<<"\n stack\n";

    qu_st_fc(q, stack, top, f, r);
    printst(stack, top);

    return 0;
}

```

Q2/B/ Having a Queue of size (10), write complete program to split the even values of the queue into other queue of size (5) and the odd values into Queue of size (5).

```
-----

#include <iostream>

using namespace std;

int const sizeq1 = 10;
int const sizeq2 = 5;
int const sizeq3 = 5;

void insertion(int q1[sizeq1], int &f1, int &r1)
{
    int value;
    int i;

    for(i = 0; i <= 9; i++)
    {
        if(r1 == sizeq1-1)
            cout<<"Queue is empty, insertion is not possible\n";

        else
        {
            cout<<"Enter any number for queue : ";
            cin>>value;

            ++r1;
            q1[r1] = value;
        }

        if(f1 == -1)
            f1 = 0;
    }
}

void spillt_qu_st(int q1[sizeq1], int q2[sizeq2], int q3 [sizeq3], int &f1, int
&r1, int &f2, int &r2, int &f3, int &r3)
{
    int i;

    for(i = r1; i >= f1; i--)
    {
        if(q1[i] % 2 == 0)
        {
            r2++;
            q2[r2] = q1[i];

            if(f2 == -1)
                f2 = 0;
        }

        else if(q1[i] % 2 != 0)
        {
            r3++;
            q3 [r3] = q1[i];
        }
    }
}
```

```

        if(f3 == -1)
            f3 = 0;
    }
}

void printq1(int q1[sizeq1], int f1, int r1)
{
    int value;
    int i;

    cout<<"\n Queue 1 full : \n";

    if(r1 == -1)
        cout<<"Queue 1 is empty\n";
    else
    {
        for(i = r1; i >= f1; i--)
            cout<<q1[i]<<"\t";
    }
}

void printq2(int q2[sizeq2], int f2, int r2)
{
    int value;
    int i;

    cout<<"\n Queue 2 even : \n";

    if(r2 == -1)
        cout<<"Queue 2 is empty\n";
    else
    {
        for(i = r2; i >= f2; i--)
            cout<<q2[i]<<"\t";
    }
}

void printst(int q3[sizeq3], int f3, int r3)
{
    int value;
    int i;

    cout<<"\n Queue 3 odd : \n";
    if(r3 == -1)
        cout<<"Queue 3 is empty\n";
    else
    {
        for(i = r3; i >= f3; i--)
            cout<<q3 [i]<<"\t";
    }
}

int main()
{
    int q1[sizeq1];

```

```

    int q2[sizeq2];
    int q3[sizeq3];

    int f1 = -1;
    int r1 = -1;

    int f2 = -1;
    int r2 = -1;

    int f3 = -1;
    int r3 = -1;

    insertion(q1, f1, r1);
    cout<<"\n";

    spillt_qu_st(q1, q2, q3 ,f1, r1, f2, r2, f3, r3);
    cout<<"\n";

    printq1(q1, f1, r1);
    cout<<"\n";

    printq2(q2, f2, r2);
    cout<<"\n";

    printst(q3 , f3, r3);
    cout<<"\n";

    return 0;
}

```

```

#include <iostream>

using namespace std;

int const size = 10;

void insertion(int q[size], int &f, int &r)
{
    int value;
    int i;

    for(i = 0; i <= 9; i++)
    {
        if(r == size-1)
            cout<<"Queue is empty, insertion is not possible\n";

        else
        {
            cout<<"Enter any number for queue : ";
            cin>>value;

            ++r;
            q[r] = value;
        }

        if(f == -1)
            f = 0;
    }
}

void printq(int q[size], int f, int r)
{
    int i;

    cout<<"Queue is :\n";
    if(r == -1)
        cout<<"Queue is empty\n";
    else
    {
        for(i = r; i >= f; i--)
            cout<<q[i]<<"\t";
    }
}

void con_qu_st_arr(int q[size], int stack[size], int array_even[size], int &top,
int &f, int &r, int &x)
{
    int i, j = 0;

    for(i = r; i >= f; i--)
    {
        if(q[i] %2 == 0)
        {
            array_even[j] = q[i];
            ++j;
            ++x;
        }
        else
        {

```

```

        ++top;
        stack[top] = q[i];
    }
}

void printst(int stack[size], int top)
{
    int i;
    cout<<"stack ood  is :\n";

    if(top == -1)
        cout<<"Stack is empty \n";

    else
    {
        for(i = 0; i <= top; i++)
            cout<<stack[i]<<"\t";
    }
}

void printarr(int array_even[size], int x)
{
    int i;
    cout<<"array even  is :\n";

    for(i = 0; i < x; i++)
        cout<<array_even[i]<<"\t";

}

int main()
{
    int q[size];
    int array_even[size];
    int stack[size];

    int top = -1;
    int f = -1;
    int r = -1;
    int x = 0;

    insertion(q, f, r);
    cout<<"\n";

    printq(q, f, r);
    cout<<"\n";

    con_qu_st_arr(q, stack, array_even, top, f, r, x);
    cout<<"\n";

    printarr(array_even, x);
    cout<<"\n";

    printst(stack, top);

    return 0;
}

```

```

#include <iostream>

using namespace std;

const int size = 8;

void insertion(int q[size], int &f, int &r)
{
    int value;
    int i;

    for(i = 0; i<=5; i++)
    {
        if(r == size - 1)
            cout<<"Queue is full !! Insertion is not possible.\n";

        else
        {
            cout<<"Enter any number : ";
            cin>>value;

            ++r;
            q[r] = value;
        }

        if(f == -1)
            f = 0;
    }
}

void deletion(int q[size], int &f, int &r)
{
    int value;
    int i;

    for(i = 0; i <= 2; i++)
    {
        if(f == -1)
            cout<<"Under flow !! Queue is empty deletion is not possible.\n";

        else if(f == r)
        {
            value = q[f];

            f = -1;
            r = -1;
        }

        else
        {
            value = q[f];
            ++f;
        }
    }
}

void print(int q[size], int f, int r)
{

```

```

    int i;

    if(f == -1)
        cout<<"Queue is empty nothing to print !!";

    else
    {
        for(i = r; i >= f; i--)
        {
            cout<<q[i]<<"\t";

        }
    }

}

int main()
{
    int q[size];
    int f = -1;
    int r = -1;

    insertion(q, f, r);
    cout<<"\n";
    print(q, f, r);
    cout<<"\n";
    deletion(q, f, r);
    cout<<"\n";
    print(q, f, r);
    cout<<"\n";

    return 0;
}

```