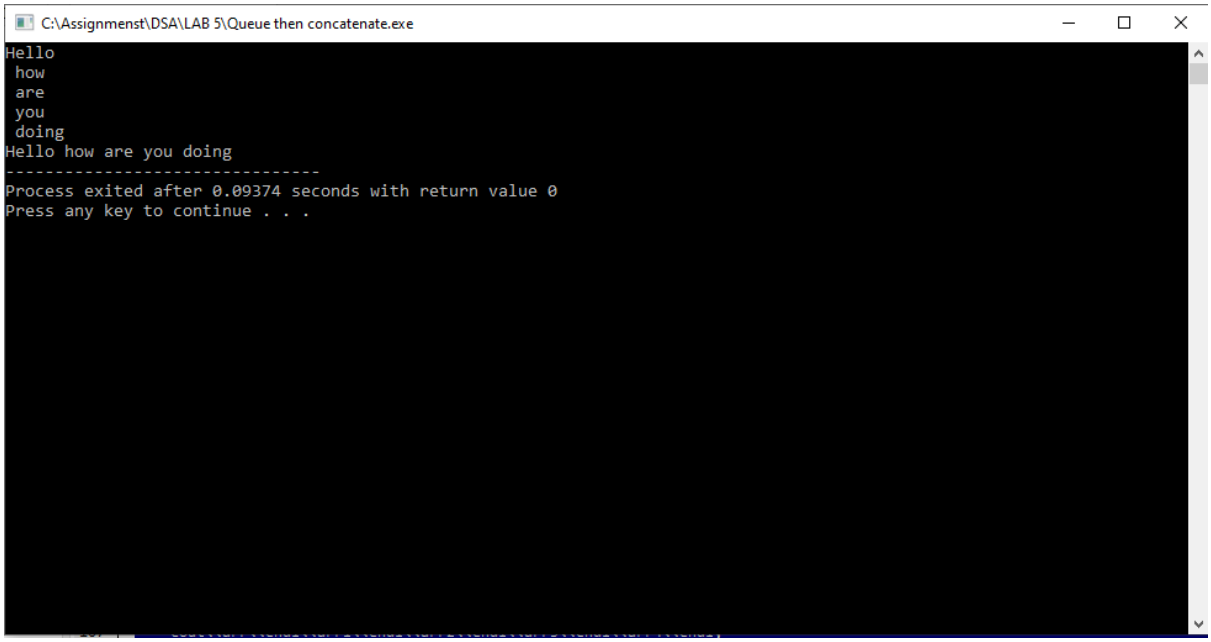
TASK 2

```
#include <iostream>
using namespace std;
class queue{
	private:
		int front=0,front1=0,front2=0,front3=0,front4=0,rear=0,rear1=0,rear2=0,rear3=0,rear4=0;
		char arr[10000],arr1[10000],arr2[10000],arr3[10000],arr4[10000];
		public:
			void enqueue(){
				cout<<"Enter the value to add to queue"<<endl;
				cin>>arr[rear];
				rear++;

			}
			void dequeue(){
			if(front==rear){
			cout<<"The queue is empty\n";
			}
			else{

			cout<<"The value removed is "<<arr[front]<<endl;
				front++;
}}
int top(){
	if(front==rear){
		cout<<"the queue is empty"<<endl;
	}
	else{

	cout<<"The value at the front of the queue is "<<arr[front]<<endl;
```

```
        return arr[front];}
}
void empty(){
        if(front==rear){
        cout<<"the queue is empty"<<endl;
        }
        else{
        cout<<"the queue exists"<<endl;
        }
}
void display(){   if(front==rear){
        cout<<"the queue is empty"<<endl;
        }
        else{
        cout<<"all the values in the queue are "<<endl;
        for (int i=front;i<rear;i++){
                cout<<arr[i]<<endl;
        }}
}
void save(string x){
        int a=0;
        for(int i=0;i<23 ;i++){
                if(x[i]==' '){
                        a++;
                }
                if(a==0){
                        arr[rear]=x[i];
                rear++;
                }
                if(a==1){
                        arr1[rear1]=x[i];
```

```
                rear1++;
                }
                if(a==2){
                        arr2[rear2]=x[i];
                rear2++;
                }
                if(a==3){
                        arr3[rear3]=x[i];
                rear3++;
                }
                if(a==4){
                        arr4[rear4]=x[i];
                rear4++;
                }

        }
}
void conc(){

        for(int i=0;i<rear1;i++){

        arr[rear]=arr1[front1];
        rear++;
        front1++;
}
        for(int i=0;i<rear2;i++){

        arr[rear]=arr2[front2];
        rear++;
        front2++;
}
```

```
        for(int i=0;i<rear3;i++){

        arr[rear]=arr3[front3];
        rear++;
        front3++;
}
        for(int i=0;i<rear4;i++){

        arr[rear]=arr4[front4];
        rear++;
        front4++;
}
cout<<arr;
}
void disp(){
        cout<<arr<<endl<<arr1<<endl<<arr2<<endl<<arr3<<endl<<arr4<<endl;
}
};
int main(){
        string ada="Hello how are you doing";
queue que;
que.save(ada);
que.disp();
que.conc();

}
```



TASK 1

```
#include <iostream>
using namespace std;
class queue{
	private:
		int front=0,rear=0,arr[10000];
		public:
			void enqueue(){
				cout<<"Enter the value to add to queue"<<endl;
				cin>>arr[rear];
				rear++;

			}
			void dequeue(){
			if(front==rear){
```

```
                cout<<"The queue is empty\n";
                }
                else{

                cout<<"The value removed is "<<arr[front]<<endl;
                        front++;
}}
int top(){
        if(front==rear){
                cout<<"the queue is empty"<<endl;
        }
        else{

        cout<<"The value at the front of the queue is "<<arr[front]<<endl;
        return arr[front];}
}
void empty(){
        if(front==rear){
        cout<<"the queue is empty"<<endl;
        }
        else{
        cout<<"the queue exists"<<endl;
        }
}
void display(){   if(front==rear){
        cout<<"the queue is empty"<<endl;
        }
        else{
        cout<<"all the values in the queue are "<<endl;
        for (int i=front;i<rear;i++){
                cout<<arr[i]<<endl;
```

```
        }}
}


};
int main(){
        queue que;
        int ch;
        int br=1;
        Label:
        cout<<"\n\nWhat would you like to do?\n1. enqueue\t2. dequeue\n3.Check the front of queue\t4.check if the queue is empty\n5.Display all the values in queue\nAny other number to exit"<<endl;
        cin>>ch;
        switch(ch){
        case 1:
        que.enqueue();
        break;
        case 2:
        que.dequeue();
        break;
        case 3:
                que.top();
                break;

        case 4:
                que.empty();
                break;
                case 5:
                        que.display();
                        break;
        default:
                br=0;
```

```
                    break;


}
if(br==1){
goto Label;
}


}
```