1\*\*\*\*\*stack implementaion

#include <iostream>

using namespace std;

#define SIZE 5

int A[SIZE];

int top = -1;

void push(int value)

{

if(top==SIZE-1)

{ cout<<"Stack is full!\n";

}

else

{

top++;

A[top]=value;

}

}

void pop()

{

if(top<0)

cout<<"Stack is empty!\n";

else

top--;

}

void print()

{

cout<<"current stack: ";

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

cout<<A[i]<<" ";

cout<<endl;

}

int main()

{

push(4);

print();

push(7);

print();

push(1);

print();

pop();

print();

push(3);

print();

push(7);

print();

pop();

print();

return 0;

}

\*\*\*2.queue implementaion

#include <iostream>

using namespace std;

#define SIZE 5

int A[SIZE];

int front = -1;

int rear = -1;

void enqueue ( int value )

{

if (rear == SIZE-1)

cout<<"Queue is full \n";

else

{

if( front == -1)

front = 0;

rear++;

A[rear] = value;

}

}

void dequeue ( )

{

if(front == -1 && rear == -1)

cout<<"Queue is empty\n";

else

if( front == rear )

front = rear = -1;

else

front++;

}

void displayQueue()

{

for( int i=front ; i<= rear ; i++)

cout<<A[i]<<" ";

cout<<"\n";

}

int main()

{

enqueue(2);

displayQueue();

enqueue(3);

displayQueue();

enqueue(5);

displayQueue();

enqueue(7);

displayQueue();

enqueue(8);

displayQueue();

enqueue(9);

displayQueue();

dequeue();

displayQueue();

dequeue();

displayQueue();

dequeue();

displayQueue();

dequeue();

displayQueue();

dequeue();

return 0;

}

\*\*\*3.circular queue

#include <iostream>

using namespace std;

#define SIZE 5

int A[SIZE];

int front = -1;

int rear = -1;

void enqueue ( int value )

{

if ((rear + 1)%SIZE == front)

cout<<"Queue is full \n";

else

{

if( front == -1)

front = 0;

rear = (rear+1)%SIZE;

A[rear] = value;

}

}

void dequeue ( )

{

if(front == -1 && rear == -1)

cout<<"Queue is empty\n";

else

if( front == rear )

front = rear = -1;

else

front = (front + 1)%SIZE;

}

void displayQueue()

{

cout<<" the queue elements are: ";

for(int i=front;i!=rear;i=(i+1)%SIZE){

cout<<A[i]<<" ";

}

cout<<endl;

}

int main()

{

enqueue(2);

displayQueue();

enqueue(3);

displayQueue();

enqueue(5);

displayQueue();

enqueue(7);

displayQueue();

enqueue(8);

displayQueue();

enqueue(9);

displayQueue();

dequeue();

displayQueue();

dequeue();

displayQueue();

dequeue();

displayQueue();

enqueue(8);

displayQueue();

enqueue(9);

displayQueue();

}

\*\*\*\*4.linear search

#include <iostream>

using namespace std;

int linearSearch(int array[], int size, int searchValue)

{

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

{

if (searchValue == array[i])

{

return i;

}

}

return -1;

}

int main()

{

int a[] = {15, 23, 7, 45, 87, 16};

int userValue;

cout << "Enter an integer: " << endl;

cin >> userValue;

int result = linearSearch(a, 6, userValue);

if(result >= 0)

{

cout << "The number " << a[result] << " was found at the"

" element with index " << result << endl;

}

else

{

cout << "The number " << userValue << " was not found. " << endl;

}

}

\*\*\*\*5.binary search

#include <iostream>

using namespace std;

int binarySearch(int array[], int size, int searchValue)

{

int start = 0;

int end = size - 1;

int mid;

while (start <= end)

{

mid = (start + end) / 2;

if(searchValue == array[mid])

{

return mid;

}

else if (searchValue > array[mid])

{

start = mid + 1;

}

else

{

end = mid - 1;

}

}

return -1;

}

int main()

{

int a[] = {12, 22, 34, 47, 55, 67, 82, 98};

int userValue;

cout << "Enter an integer: " << endl;

cin >> userValue;

int result = binarySearch(a, 8, userValue);

if(result >= 0)

{

cout << "The number " << a[result] << " was found at the"

" element with index " << result << endl;

}

else

{

cout << "The number " << userValue << " was not found. " << endl;

}

}

\*\*\*6.bubble sorting

#include<iostream>

using namespace std;

int main()

{

int n, arr[50], temp;

cout<<"Enter total number of elements :";

cin>>n;

cout<<"Enter "<<n<<" numbers :";

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

{

cin>>arr[i];

}

cout<<"Sorting array using bubble sort technique...\n";

for(int i=0; i<(n-1); i++)

{

for(int j=0; j<(n-i-1); j++)

{

if(arr[j]>arr[j+1])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

cout<<"Sorted list in ascending order :\n";

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

{

cout<<arr[i]<<" ";

}

}