**Assignment-40: A Job Ready Bootcamp in c++,DSA and IOT**

**deque**

1. Inserts an element. And returns an iterator that points to the first of the newly inserted

elements.

Sol – 1.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

deque<int>::iterator it=dq.begin();

it++;

it=dq.insert(it,10);

cout<<"Deque : ";

for(it=dq.begin();it!=dq.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

2. Returns a reverse iterator which points to the last element of the deque (i.e., its

reverse beginning).

Sol – 2.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

for(auto it=dq.rbegin();it!=dq.rend();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

3. Returns a reverse iterator which points to the position before the beginning of the

deque (which is considered its reverse end).

Sol – 3.

Same as previous

4. Returns a constant iterator pointing to the first element of the container, that is, the

iterator cannot be used to modify, only traverse the deque.

Sol – 4.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

deque<int>::const\_iterator it;

for(it=dq.begin();it!=dq.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

5. Returns the maximum number of elements that a deque container can hold.

Sol – 5.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

cout<<"Max element deque can hold : "<<dq.max\_size();

cout<<endl;

return 0;

}

6. Assign values to the same or different deque container.

Sol – 6.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq;

dq.assign(5,10);

cout<<"Elements of dq : ";

for(auto it=dq.begin();it!=dq.end();it++)

cout<<\*it<<" ";

cout<<endl;

deque<int>dq2;

dq2.assign(dq.begin(),dq.end());

cout<<"Elements of dq2 : ";

for(auto it=dq2.begin();it!=dq2.end();it++)

cout<<\*it<<" ";

cout<<endl;

return 0;

}

7. Return the first element and last element of the deque container.

Sol – 7.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

cout<<"Front element : "<<dq.front()<<endl;

cout<<"Back element : "<<dq.back()<<endl;

return 0;

}

8. Remove elements from a container from the specified position or range in deque.

Sol – 8.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

deque<int>::iterator it=dq.begin();

dq.erase(++it);

for(auto it=dq.begin();it!=dq.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

9. Generate a permutation of first N natural numbers having count of unique adjacent

differences equal to K | Set 2 using a queue.

Sol – 9.

#include<bits/stdc++.h>

using namespace std;

void permutation(int n,int k)

{

deque<int>dq;

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

{

dq.push\_back(i);

}

int front=1;

vector<int>ans;

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

{

if(front==1)

{

ans.push\_back(dq.front());

dq.pop\_front();

if(k>1)

{

front=0;

}

k--;

}

else

{

ans.push\_back(dq.back());

dq.pop\_back();

if(k>1)

{

front=1;

}

k--;

}

}

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

cout<<ans[i]<<" ";

cout<<endl;

}

int main()

{

permutation(5,3);

return 0;

}

10. Check if given Strings can be made equal by inserting at most 1 String using deque.

Sol – 10.

#include<bits/stdc++.h>

using namespace std;

bool aresimilar(string s1,string s2)

{

int n=s1.size();

int m=s2.size();

if(n==m)

{

if(s1==s2)

return true;

return false;

}

deque<string>x,y;

s1.push\_back(' ');

s2.push\_back(' ');

string temp="";

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

{

if(s1[i]!=' ')

{

temp=temp+s1[i];

}

else{

x.push\_back(temp);

temp="";

}

}

for(int i=0;i<m+1;i++)

{

if(s2[i]!=' ')

{

temp=temp+s2[i];

}

else{

y.push\_back(temp);

temp="";

}

}

while(x.size()>0&&y.size()>0&&x.front()==y.front())

{

x.pop\_front();

y.pop\_front();

}

while(x.size()>0&&y.size()>0&&x.back()==y.back())

{

x.pop\_back();

y.pop\_back();

}

if(x.size()==0||y.size()==0)

return true;

return false;

}

int main()

{

string S1="My name is Gurudev Singla";

string S2="My name Singla";

if(aresimilar(S1,S2))

{

cout<<"True";

}

else

cout<<"False";

return 0;

}

11. How to get the first and last elements of Deque in c++?

Sol – 11.

#include<bits/stdc++.h>

using namespace std;

int main()

{

deque<int>dq={1,2,5,6};

cout<<"Front : "<<dq.front()<<endl;

cout<<"Back : "<<dq.back()<<endl;

return 0;

}

12. Given a string S containing letters and ‘#‘. The ‘#” represents a backspace. The task

is to print the new string without ‘#‘. String after processing backspace characters

using deque

Examples:

Input : S = "abc#de#f#ghi#jklmn#op#"

Output : abdghjklmo

Input : S = "##iNeuron##Education##hub#"

Output : iNeurEducatihu

Sol – 12.

#include<bits/stdc++.h>

using namespace std;

void ns(string s)

{

deque<char>dq;

for(int i=0;i<s.length();i++)

{

if(s[i]=='#')

{

if(dq.empty())

continue;

dq.pop\_back();

}

else

dq.push\_back(s[i]);

}

cout<<"Dequeue : ";

for(auto it=dq.begin();it!=dq.end();it++)

{

cout<<\*it;

}

cout<<endl;

}

int main()

{

string s="abc#de#f#ghi#jklmn#op#";

ns(s);

return 0;

}

13. Segregate even and odd nodes in a Linked List using Deque.

Sol – 13.

void evenodd(struct Node\* head)

{

struct Node\* temp=head;

deque<int>dq;

while(temp!=NULL)

{

if(temp->data%2==0)

dq.push\_front(temp->data);

else

dq.push\_back(temp->data);

temp=temp->next;

}

}