/\*

A palindrome is a string of character thates the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, .Poor Dan is in a droop. is a palindrome, as can be seen by examining the characters \poor danisina droop. and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions

1. to check whether given string is palindrome or not that uses a stack to determine whether a string is a palindrome.

2. to remove spaces and punctuation in string, convert all the Characters to lowercase, and then call above Palindrome checking function to check for a palindrome

3. to print string in reverse order using stack

\*/

#include<iostream>

#include<string.h>

#define MAX 100

using namespace std;

typedef struct stack

{

char data[MAX];

int top;

}stack;

void init(stack \*stck)

{

int i;

for(i=0;i<MAX;i++)

stck->data[i]='\0';

stck->top=-1;

}

void print(stack stck)

{

int i;

cout<<"\nStack element are::";

for(i=0;i<MAX;i++)

cout<<stck.data[i];

cout<<"\ttop="<<stck.top;

}

int isempty(stack stck)

{

return stck.top==-1?1:0;

}

int isfull(stack stck)

{

return stck.top==MAX-1?1:0;

}

void push(stack \*stck,char data[MAX])

{

int i;

for(i=0;data[i]!='\0';i++)

{

stck->top+=1;

stck->data[stck->top]=data[i];

}

}

void pop(stack \*stck)

{

char data[MAX],data1[MAX];

int i=0;

while(!isempty(\*stck))

{

data[i]=stck->data[stck->top];

stck->top-=1;

i++;

}

data[i]='\0';

cout<<"\nOriginal String is="<<stck->data;

cout<<"\nReversed string is::";

cout<<data;

if(strcmp(data,stck->data))

cout<<"\nString is not palindrome";

else

cout<<"\nString is plindrome";

}

int main()

{

stack stck;

char data[MAX]={'P','o','o','r',' ','D','a','n',' ','i','s',' ','a',' ','d','r','o','o','p','\0'};

char data1[MAX];

init(&stck);

int ch,i,j=0;

for(i=0;data[i]!='\0';i++)

{

if(data[i]!=' ')

{

if(data[i]>=65&&data[i]<=90)

data1[j]=data[i]+32;

else

data1[j]=data[i];

j++;

}

}

data1[j]='\0';

while(ch!=3)

{

cout<<"\n\*\*\*\*\*\*Stack Demo \*\*\*\*\*\*\*\n";

cout<<"\n1.Push the string,print reverse string & check palindrome or not(pop) ::";

cout<<"\n2.Remove blank space,convert uppercase into lowercase,push the string & check palindrome or not (pop)";

cout<<"3.Exit";

cout<<"\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

if(isfull(stck))

cout<<"\nstack is full";

else

{

push(&stck,data);

}

if(isempty(stck))

cout<<"\nStack is empty::";

else

pop(&stck);

break;

case 2:

init(&stck);

if(isfull(stck))

cout<<"\nStack is full";

else

{

push(&stck,data1);

}

pop(&stck);

break;

}

}

return 0;

}

/\* OUTPUT:

\*\*\*\*\*\*Stack Demo \*\*\*\*\*\*\*

1.Push the string,print reverse string & check palindrome or not(pop) ::

2.Remove blank space,convert uppercase into lowercase,push the string & check palindrome or not (pop)3.Exit

Enter your choice ::1

Original String is=Poor Dan is a droop

Reversed string is::poord a si naD rooP

String is not palindrome

\*\*\*\*\*\*Stack Demo \*\*\*\*\*\*\*

1.Push the string,print reverse string & check palindrome or not(pop) ::

2.Remove blank space,convert uppercase into lowercase,push the string & check palindrome or not (pop)3.Exit

Enter your choice ::2

Original String is=poordanisadroop

Reversed string is::poordasinadroop

String is not palindrome

\*\*\*\*\*\*Stack Demo \*\*\*\*\*\*\*

1.Push the string,print reverse string & check palindrome or not(pop) ::

2.Remove blank space,convert uppercase into lowercase,push the string & check palindrome or not (pop)3.Exit

Enter your choice ::

\*/