**Sri Sri University, Cuttack, Odisha.**

**Faculty of Science**

|  |  |
| --- | --- |
| **Program: B.Sc. – Computer Science, Data Science, & Environmental Science**  **(2020-23 Batch)**  **Subject Code/Subject Name: Data Structure Laboratory**  **Assignment –VI** | |
| **Full Name of the Student:** | VINAYAK SANJAY CHAVAN |
| **Full Roll Number:** | BCS-011 |
| **Program:** | B.Sc. (Computer Sc.) / B.Sc. (Data Sc.) / B.Sc. (Env. Sc.) |
| **Date:** | 22nd March, 2021 (10.00 AM – 12.00 Noon) |
| **Signature** |  |

|  |  |
| --- | --- |
| **All Questions are compulsory** | **Total Marks: 50** |

|  |  |
| --- | --- |
| **Question (s)** | **Maximum Marks** |
| 1. Write a structure for an double stack, implement function push(), pop(), and peek(), IsEmpty() and IsFull() function. Write a main function and call the functions based on an option entered.   **Your code:** **/\* Write a structure for an double stack, implement function**  **push(), pop(), and peek(), IsEmpty() and IsFull() function. Write**  **a main function and call the functions based on an option entered \*/**  **#include <iostream>**  **using namespace std;**  **int stack[5], n=100, top=-1;**  **void push(int val){**  **cout<<"stack"<<endl;**  **top++;**  **stack[top]=val;**  **}**  **void pop(){**  **/\* if(top<=-1)**  **cout<<"stack"<<endl;**  **else {\*/**  **cout<<"popped element is "<< stack[top] <<endl;**  **top--;**  **// }**  **}**  **void display(){**  **if(top>=0){**  **cout<<"stack elements are:";**  **for(int i=top; i>=0; i--){**  **cout<<stack[i]<<" ";**    **}**  **}else**  **cout<<"stack is empty";**  **cout<<endl;**  **}**  **void isEmpty(){**  **if(top==-1){**  **cout<<"stack is empty"<<endl;**  **}else**  **cout<<"stack is not empty"<<endl;**  **}**  **void isFull(){**  **if(top == n - 1){**  **cout<<"stack is full"<<endl;**  **}else**  **cout<<"stack is not full"<<endl;**  **}**  **int peek(int element){ //void**  **return stack[top];**  **}**  **int main(){**  **int ch, val;**  **do{**  **cout<<"1. Push in the stack "<<endl;**  **cout<<"2. Pop from the stack "<<endl;**  **cout<<"3. Display the stack "<<endl;**  **cout<<"4. stack is empty "<<endl;**  **cout<<"5. stack is full "<<endl;**  **cout<<"6. peek "<<endl;**  **cout<<"7. Exit"<<endl;**  **cout<<"Enter choice: "<<endl;**  **cin>>ch;**  **switch(ch){**  **case 1:{**  **cout<<"enter value to be pushed:"<<endl;**  **cin>>val;**  **push(val);**  **break;**  **}**  **case 2:{**  **pop();**  **break;**  **}**  **case 3:{**  **display();**  **break;**  **}**  **case 4:{**  **isEmpty();**  **break;**  **}**  **case 5:{**  **isFull();**  **break;**  **}**  **case 6:{**  **int element;**  **cout<<"the element is: "<<peek(element);**  **peek(element);**  **break;**  **}**  **case 7:{**  **cout<<"exit"<<endl;**  **break;**  **}**  **default:{**  **cout<<"invalid Choice"<<endl;**  **}**  **}**  **}**  **while(ch!=6);**  **return 0;**  **}**  **Screenshot of output:** | 20 |
| 1. Implement a stack of characters and create **mystack.h**. Write a program to check whether an entered string is a palindrome or not. You need to include mystack.h for doing this.     **Your code:**  **#include <iostream>**  **using namespace std;**  **#define size 20**  **typedef struct stack{**  **char array[size];**  **int top;**  **}STACK;**  **char push(STACK\*,char);**  **char pop(STACK\*);**  **char peek(STACK\*);**  **bool full\_stack(STACK\*);**  **int empty\_stack(STACK\*);**  **void display(STACK\*);**  **void stack\_initialisation(stack \*s1){**  **s1->top=-1;**  **}**  **char push(stack \*s1,char element ){**  **s1->top++;**    **s1->array[s1->top]=element;**  **return element;**  **}**  **char pop(stack \*s1){**  **double element;**  **element=s1->array[s1->top];**  **s1->top--;**  **return element;**  **}**  **char peek(stack \*s1){**  **return s1->array[s1->top];**  **}**  **bool full\_stack(stack \*s1){**  **if(s1->top==size-1){**  **cout<<"yes stack is full "<<endl;**  **return true;**  **}**  **else{**  **cout<<"stack has space "<<endl;**  **return false;**  **}**  **}**  **int empty\_stack(stack \*s1){**  **if(s1->top==-1){**  **return 1;**  **}**  **else{**  **return 0;**  **}**  **}**  **void display(stack \*s1) {**  **if(s1->top >-1) {**  **cout<<"Stack elements are:";**  **for(int i=s1->top; i>=0; i--)**  **cout<<s1->array[i]<<" ";**  **cout<<endl;**  **}else**  **cout<<"Stack is empty";**  **}**  **#include<iostream>**  **#include <stdlib.h>**  **#include <string.h>**  **#include"stack.h"**  **using namespace std;**  **#define size 20**  **int main(){**  **STACK s;**  **int i=0,j=0;**  **char ans[20],element;**  **int length;**  **int palindrome;**  **stack\_initialisation(&s);**  **cout<<"1 check the palindrome "<<endl;**  **cout<<"2 exit"<<endl;**  **int option;**  **cout<<"choose option "<<endl;**  **cin>>option;**  **switch(option){**  **case 1:**  **cout<<"Enter the string: ";**  **cin>>s.array;**  **length=strlen(s.array);**    **while(i<length){**  **push(&s,s.array[i]);**  **i++;**  **}**    **while(j<length){**  **element=pop(&s);**  **ans[j]=element;**  **j++;**  **}**  **for(int j=0;j<length;j++){**  **cout<<ans[j];**  **}**  **cout<<endl;**  **palindrome=strcmp(s.array,ans);**  **if(palindrome==0){**  **cout<<"The string is palindrome.";**  **}else{**  **cout<<"The string is not palindrome.";**  **}**  **break;**  **case 2:**  **exit(0);**  **}**  **}**  **Screenshot of output:** | 20 |
| 1. Write a program to convert decimal to binary using an integer stack.   Your code:  #include<iostream>  #define size 100  using namespace std;  typedef struct stack{  int arr[size];  int top;  }STACK;  void stack\_inti(STACK\*);  void push(STACK\*,int);  int pop(STACK\*);  void dec\_bin(int);  int main(){  int num;  cout<<"Enter the decimal value: ";  cin>>num;  cout<<"The binary value of the decimal number is: ";  dec\_bin(num);  }  void stack\_inti(STACK \*s1){  s1->top=-1;  }  void push(STACK \*s1,int n){  s1->top++;  s1->arr[s1->top]=n;  }  int pop(STACK \*s1){  int element;  element=s1->arr[s1->top];  s1->top--;  return element;  }  void dec\_bin(int ele){  int rem;  STACK s;  stack\_inti(&s);  while(ele>0){  rem=ele%2;  push(&s,rem);  ele=ele/2;  }  for(int i=s.top;i!=-1;i--){  cout<<pop(&s);  }  }  Screenshot of output: | 10 |