**Assignment Submission**

**Assignment : 2**

**Student ID: 24BCA095**

|  |  |
| --- | --- |
| **1** | **Write a program to perform the following operations on a stack. (Implement the stack using array)**  **• PUSH • POP • PEEP • CHANGE** |
|  | #include <iostream>  using namespace std;  #define MAX 5  int TOP = -1 ,s[MAX], POS, ELE;  void push(){      if (TOP == MAX - 1) {          cout << "Stack Overflow\n";          return;      }      else{          TOP++;          cout << "Enter the element to push: ";          cin >> s[TOP];          cout << "Element pushed: " << s[TOP] << " at index: "<<TOP  << endl;      }  }  void pop(){      if (TOP == -1 ){          cout << "Stack Underflow\n";          return;      }      else {          cout << "Element popped: " << s[TOP] << " from index: " << TOP << endl;          TOP--;      }  }  int peep(int POS){      if ( TOP - POS +1 <= 0){          cout << "Invalid position\n";          return -1;      }      else {          return s[TOP - POS + 1];      }  }  void change(int POS){      cout << "Enter element to change at position " << TOP - POS + 1 << ": ";      cin >> ELE;      if (TOP - POS + 1 <= 0) {          cout << "Invalid position for change\n";          return;      }      else {          s[TOP - POS + 1] = ELE;          cout << "Element at position " << TOP - POS + 1 << " changed to: " << ELE << endl;      }  }  int main(){      while(1){          cout << "1. Push\n2. Pop\n3. Peep\n4. Change\n5. Exit\n";          int ch;          cin >> ch;          switch(ch){              case 1: push(); break;              case 2: pop(); break;              case 3:                  cout << "Enter position to peep ";                  cin >> POS;                  {                  int ans = peep(POS);                  if (ans == -1)                      cout << "Pop operation failed.\n";                  else                      cout << "Element at postion " << POS << " is: " << ans << endl;                  }                      break;              case 4:                  cout << "Enter position to change : ";                  cin >> POS;                  change(POS);                  break;              case 5: return 0;              default: cout << "Invalid choice, please try again.\n";          }      }  } |
| **2** | **Write a program to perform the following operation on a simple queue. (Implement the queue using array) • Insert an element • Remove an element.** |
|  | #include <iostream>  using namespace std;  #define MAX 10  int FRONT = -1 , REAR = -1 ,ELE,Q[MAX];  void insertion(){      if (REAR == MAX){          cout << "Queue is full\n";          return;      }      else {          cout << "Enter value to insert at index " << REAR << ": ";          cin >> ELE;          REAR ++ ;          Q[REAR] = ELE;          if (FRONT == -1 ){              FRONT = 0;          }          return;      }  }  void deletion(){      if (FRONT == -1 ){          cout << "Queue is empty\n";          return;      }      else {          ELE = Q[FRONT];          if (FRONT == REAR ){              FRONT = -1;              REAR = -1;          }          else {              FRONT ++;          }          cout << "Deleted element: " << ELE << " at index: " << FRONT << endl;          return;      }  }  void display(){      if ( FRONT == -1 ){          cout << "Queue is empty\n";          return;      }      else {          for (int i = FRONT; i <= REAR; i++)              cout << Q[i] << " ";          cout << endl;          return;      }  }  int main(){      while (1){          cout << "1. Insertion\n2. Deletion\n3. Display\n4. Exit\n";          int choice;          cin >> choice;          switch (choice) {              case 1:                  insertion();                  break;              case 2:                  deletion();                  break;              case 3:                  display();                  break;              case 4:                  exit(0);              default:                  cout << "Invalid choice. Please try again.\n";          }      }  } |
| **3** | **Write a program to perform the following operation on a circular queue. (implement the circular queue using array) • Insert an element • Remove an element** |
|  | **#include <iostream>**  **using namespace std;**  **#define MAX 5**  **int FRONT = -1 , REAR = -1 ,ELE,Q[MAX];**  **void insertion(){**  **if ((FRONT == 0 && REAR == MAX - 1) || (FRONT == REAR + 1)) {**  **cout << "Queue is full\n";**  **return;**  **}**  **else {**  **cout << "Enter value to add : " ;**  **cin >> ELE;**  **if (REAR == MAX - 1)**  **REAR=0;**  **else if (REAR == -1 )**  **FRONT = REAR = 0;**  **else**  **REAR++;**  **Q[REAR] = ELE;**  **return;**  **}**  **}**  **void deletion(){**  **if (FRONT == -1 ){**  **cout << "Queue is empty\n";**  **return;**  **}**  **else {**  **ELE = Q[FRONT];**  **if (FRONT == REAR ){**  **FRONT = REAR = -1;**  **}**  **else if (FRONT == MAX -1 )**  **FRONT = 0;**  **else**  **FRONT++;**  **cout << "Deleted element: " << ELE << " at index: " << FRONT << endl;**  **return;**  **}**  **}**  **void display(){**  **for (int i = 0; i< MAX ; i++){**  **cout << Q[i] << " "   ;**  **}**  **cout << endl;**  **}**  **int main(){**  **while (1){**  **cout << "1. Insertion\n2. Deletion\n3. Display\n4. Exit\n";**  **int ch;**  **cin >> ch;**  **switch (ch) {**  **case 1:**  **insertion();**  **break;**  **case 2:**  **deletion();**  **break;**  **case 3:**  **display();**  **break;**  **case 4:**  **return 0;**  **default:**  **cout << "Invalid choice. Please try again.\n";**  **}**  **}**  **}** |
| **4** | **Write a program to calculate factorial of a given number using a recursion function.** |
|  | #include <iostream>  using namespace std;  int factorial(int n) {      if (n == 0)          return 1;      else          return n \* factorial(n - 1);  }  int main() {      int num;      cout << "Enter a positive integer: ";      cin >> num;      cout << "Factorial of " << num << " = " << factorial(num) << endl;      return 0;  } |
| **5** | **Write a program to find the GCD (Greatest Common Division) of two numbers using a recursion function** |
|  | #include <iostream>  using namespace std;  int gcd(int a , int b){      if (b!=0){      int temp = b;      b=a%b;      a = temp;      return gcd(a, b);  }      return a;  }  int main(){      int a,b;      cout << "Enter 2 integers : \n";      cin >> a >> b;      if(a>b)      cout << "GCD is : " << gcd(a,b) << endl;      else      cout << "GCD is : " << gcd(b,a)<< endl;  } |