**DATA STRUCTURES**

Salanatin, Nathaly Pearl F. Fernando Renegado

BSCS-NS-2A November 11, 2022

**PRELIMS**

Instruction: Write the codesin C++.Define all functions that may be used in your functions/solutions.Student Record refers to name(as key), quiz1,quiz2,quiz3.(10 points per problem)

|  |  |
| --- | --- |
| **NUMBER** | **SOURCE CODE** |
| 1.Write the function insert(string name)that inserts a unique Student Record using name as key in a sorted listusing array of structuresADT List. | void insert (machine)  {  Student\_Reco list[5];  int i;  for(i = 0; i < 5; i++)  {  if(name > list[i].name)  {  if (list[i] == NULL)  list[i] = new\_student\_rec;  else  list[i+1] = list[i];  list[i] = new\_student.rec  }  }  } |
| 2.Write the function insert(string name)that inserts a unique Student Record using name as key in a sorted listusing pointer implementation of ADT List. | void add(string name)  {  NODE \*tail, \*current, \*new\_node;  tail = current = Head;  new\_node = (NODE\*) ((malloc(sizeof(NODE))));  new\_node -> stu\_data = input;  // sort algorithm  while(tail != NULL && strcmp(tail->stu\_data.name,name) > 0)  {  current = tail;  tail = tail->link;  }  // add algorithm  if(tail==Head)  {  Head = new\_node;  }  else  {  current = new\_node;  }  new\_node->link = tail;  } |
| 3.Write the function update(string name) that updates a Student Record using arrayof structures implementation of ADT List. | void update(string name)  {  Student\_Reco list[5];  int i = position\_to\_update;  list[i].name = name;  } |
| 4.Write function enqueue(int x) in a pointer implementation of Queue of Integers. | void enQueue(int x)  {  QNode\* temp = new QNode(x);  if (rear == NULL)  {  front = rear = temp;  return;  }  rear->next = temp;  rear = temp;  } |
| 5.Write function pop() in a pointer implementation of Stack of Integers. | int pop()  {  if(top<0)  {  cout<<"Error! Stack is empty… \n";  return INT\_MIN;  }  else  {  int x=elements[top--];  return x;  }  } |
| 6.Write function display() inan array implementation of Queue of integers. | void display()  {  int i;  if(isEmpty())  cout << endl << "Error! Queue is empty!" << endl;  else  {  cout << endl << "Queue Elements : ";  for(i = front; i <= rear; i++)  cout << myQueue[i] << "\t";  }  } |
| 7.Write function dequeue() in an array implementation of Queue of Integers. | void Algorithm::dequeue()  {  if(isEmpty()) {  system("cls");  cout << " Error! Queue is empty!" << endl;  system("pause");  }  else {  q.front = (q.front + 1) % MAX;  q.count--;  }  } |
| 8.Write function merge (L1,L2) that merges L2 into L1. L1 and L2 are two sorted Link Lists of Student Records. | Node \* mergeLists(Node \* list1, Node \* list2)  {  Node \* head = new Node(-1);  Node \* temp;  temp = head;    while(list1!= nullptr && list2!= nullptr)  {  if(list1->val < list2->val)  {  temp -> next = new Node(l1->val);  temp = temp -> next;  list1 = list1 -> next;  }  else  {  temp -> next = new Node(list2->val);  temp = temp -> next;  list2 = list2 -> next;  }  }  while(list1!=nullptr)  {  temp -> next = new Node(list1->val);  temp = temp -> next;  list1 = list1 -> next;  }  while(list2!=nullptr)  {  temp -> next = new Node(list2->val);  temp = temp -> next;  list2 = list2 -> next;  }  return head->next;  } |
| 9.Write function purge() that deletes replicate records in a Link List of student records. | void purge(struct node\* head)  {  struct node\* current = head;  struct node\* next\_next;  if (current == NULL)  return;  while (current -> next != NULL)  {  if (current -> data == current -> next -> data)  {  next\_next = current -> next -> next;  free(current -> next);  current -> next = next\_next;  }  else  {  current = current -> next;  }  }  } |
| 10.Write function withdraw(float amount,float balance) that checksif a given amount is a valid in an ATM Withdraw transaction. | void withdraw(float amount, float balance)  {  int no\_bills\_coins\_lower\_100 = (int)amount % 100;  if(amount > balance && no\_bills\_coins\_lower\_100 > 0)  {  cout << "withdraw amount exceeds your account balance";  }  else  {  withdrawal\_transaction\_menu();  }  } |