# International Islamic University, Islamabad (IIUI) Faculty of Basic and Applied Sciences (FBAS) Data Structures & Algorithms

**Department of Computer Science**

**Lab Assignment No. 5, June 2020 [Total Marks 20]**

# Instructor: Samreen Ishfaq BSCS F18

Student Name: Mobeena Ramzan Student Name: Noor-ul-Huda

Roll No: 3868-FBAS/BSCS/F18 Roll No: 3847-FBAS/BSCS/F18

Student Name: Tashfeen Ahmed Student Name: Laiba bint Mazhar Student Name: Tehreem Fatima

Roll No: 3869-FBAS/BSCS/F18 Roll No: 3870-FBAS/BSCS/F18 Roll No: 3872-FBAS/BSCS/F18

**This assignment is due on 15th June, 2020. This page will be the front page of your assignment, you have to write your group member names and registration numbers on it. Late submission is not acceptable.**

Write a program to implement Priority Queue. Your program must include following operations:

|  |  |
| --- | --- |
| 1. Enqueue | 3 |
| 2. Dequeue | 3 |
| 3. Front | 1 |
| 4. Isempty | 1 |
| 5. Isfull (If you are using arrays) | 1 |
| 6. Find | 2 |
| 7. Size | 1 |
| 8. Display | 2 |
| 9. Exit | 1 |
| 10. Demo | 5 |

|  |
| --- |
| **Node.h**  #pragma once  #include<iostream>  #include<string>  using namespace std;  class node  {  string object;  int priority;  node \*next;  public:  node()  {  object = " ";  priority = 0;  next = nullptr;  }  ~node()  {  }  string get()  {  return object;  }  void set(string object)  {  this->object = object;  }  int getpri()  {  return priority;  }  void setpri(int priority)  {  this->priority = priority;  }  node \*getnext()  {  return next;  }  void setnext(node \*next)  {  this->next = next;  }  }; |

|  |
| --- |
| **Queue.h**  #pragma once  #include<iostream>  #include<string>  #include"node.h"  using namespace std;  class queue  {  int size;  node \*rear;  node \*front;  public:  queue()  {  size = 0;  front = nullptr;  rear = nullptr;  }  ~queue()  {  }  void enqueue();  void dequeue();  void frontq();  bool isempty();  int find();  void sizeq();  void display();  void exit();  }; |

|  |
| --- |
| **Source.cpp**  #include "queue.h"  #include "node.h"  #include<iostream>  #include<string>  using namespace std;  void queue::enqueue()  {  int x;  string n;  cout << "enter name: ";  getline(cin, n, '.');  cout << "enter priority of the name: ";  cin >> x;  node \*newnode = new node();  newnode->set(n);  newnode->setpri(x);  newnode->setnext(nullptr);  size++;  if (front == nullptr)  {  rear = newnode;  front = newnode;  }  else if (newnode->getpri() >= rear->getpri())  {  rear->setnext(newnode);  rear = newnode;  }  else if (newnode->getpri() < front->getpri())  {  newnode->setnext(front);  front = newnode;  }  else  {  node \*ltemp;  node \*temp;  ltemp = front;  temp = front;  while ((temp->getnext() != nullptr) && (newnode->getpri() >= temp->getpri()))  {  ltemp = temp;  temp = temp->getnext();  }  newnode->setnext(temp);  ltemp->setnext(newnode);  temp = newnode;  }  }  void queue::dequeue()  {  if (front != nullptr)  {  int x;  string n;  n = front->get();  x = front->getpri();  node \*p = front;  front = front->getnext();  delete p;  size--;  cout << "the name " << n << " the name priority " << x << " has been deleted" << endl;  }  else  {  cout << "the queue is empty" << endl;  }  }  void queue::frontq()  {  if (front != nullptr)  {  cout << "the front name is " << front->get() << " and its priority is " << front->getpri() << endl;  }  else  {  cout << "the queue is empty" << endl;  }  }  bool queue::isempty()  {  if (front == nullptr)  {  return true;  }  else  {  return false;  }  }  int queue::find()  {  if (front != nullptr)  {  string n;  int flag = 0;  cout << "enter the name to find: ";  getline(cin, n, '.');  node \*temp = front;  while (temp != nullptr)  {  if (n == temp->get())  {  flag ++;  cout << "name: " << temp->get() << endl  << "priority: " << temp->getpri() << endl;  break;  }  else  {  flag = 0;  temp = temp->getnext();  }  }  return flag;  }  else  {  cout << "the queue is empty" << endl;  return -1;  }  }  void queue::sizeq()  {  if (front != nullptr)  {  cout << "the size of queue is: " << size << endl;  }  else  {  cout << "the queue is empty" << endl;  }  }  void queue::display()  {  if (front != nullptr)  {  node \*temp;  temp = front;  while (temp != nullptr)  {  cout << "name is: " << temp->get() << endl;  cout << "priority is: " << temp->getpri() << endl << endl;  temp = temp->getnext();  }  }  else  {  cout << "the queue is empty" << endl;  }  }  void queue::exit()  {  if (front != nullptr)  {  cout << "program ended" << endl;  }  else  {  cout << "the queue is empty" << endl;  }  } |

|  |
| --- |
| **Queue.cpp**  #include "queue.h"  #include<iostream>  #include<string>  using namespace std;  int main()  {  queue q;  int f;  int option;  do  {  cout << endl  << "---- MENU ----" << endl  << "1: Enqueue" << endl  << "2: Dequeue" << endl  << "3: front" << endl  << "4: isempty" << endl  << "5: Find" << endl  << "6: Size" << endl  << "7: Display" << endl  << "8: Exit" << endl  << "enter your choice: ";  cin >> option;  switch (option)  {  case 1:  {  q.enqueue();  break;  }  case 2:  {  q.dequeue();  break;  }  case 3:  {  q.frontq();  break;  }  case 4:  {  if (q.isempty())  {  cout << "the queue is empty" << endl;  }  else  {  cout << "queue is not empty" << endl;  }  break;  }  case 5:  {  f = q.find();  if (f == 0)  {  cout << "name not found" << endl;  }  else  {  cout << "name found" << endl;  }  break;  }  case 6:  {  q.sizeq();  break;  }  case 7:  {  q.display();  break;  }  case 8:  {  q.exit();  break;  }  default:cout << "Error: Enter correct option from menu" << endl;  }  } while (option != 8);  system("pause");  return 0;  } |

|  |
| --- |
| **Results**: |