****

**PAMANTASAN NG LUNGSOD NG MAYNILA**

College of Information Systems and Technology Management (CISTM)

**ICC 0104-1 – Data Structures and Algorithms**

*A.Y. 2023- 2024*

**Group 4**: Searching

**Submitted by**:

Abundo, Jonalene Ryza B.

Dela Peña, Daniella Mae N.

Diamzon, Momer Ailes M.

Lau, Trisha Mae R.

Mahusay, Lindsay G.

Matanga, Sophia Vien V.

Rivera, Ramyll C.

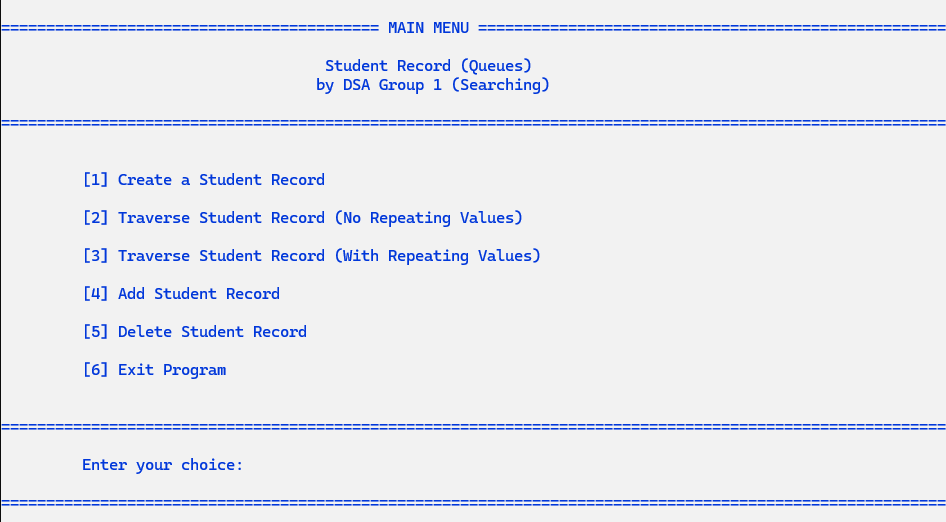
Sibayan, Joan F.

BSCS 1-1

**C++ Source Code**:

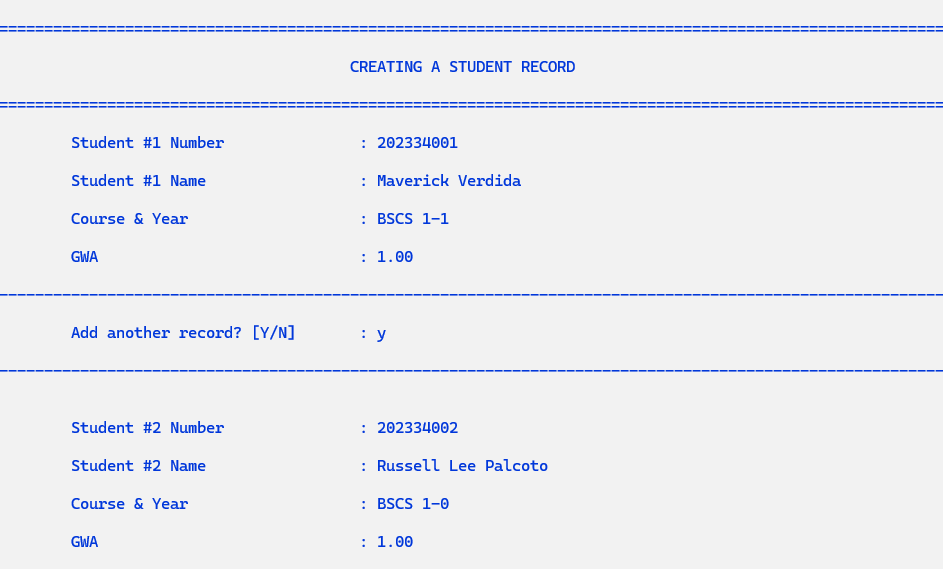
| #include <iostream>  #include <iomanip>  #include <cstdlib>  #include <cstring>  using namespace std;  struct SRec {  int SNum;  char SName[25];  char crsYr[15];  float GWA;  SRec \*next;  };  void MainMenu();  void Create(SRec \*\*front, SRec \*\*rear);  void travqnorep(SRec \*\*front, SRec \*\*rear);  void travqwrep(SRec \*\*front, SRec \*\*rear);  void addq(SRec \*\*front, SRec \*\*rear);  void delq(SRec \*\*front);  void gotoxy(int x, int y) {  cout << "\033[" << y << ";" << x << "f";  }  int main() {  SRec \*front = NULL , \*rear = NULL ;  int chc, chc2;  char temp;  menu:  MainMenu();  gotoxy(0, 27);  cout << "=========================================================================================================";  gotoxy(10, 25);  cout << "Enter your choice: ";  cin >> chc;  cin.ignore();  switch (chc) {  case 1:  Create(&front, &rear);  goto menu;  break;  case 2:  travqnorep(&front, &rear);  cout << "\n---------------------------------------------------------------------------------------------------------\n\n";  cout << "\tPress any key to return to main menu...";  cout << "\n\n---------------------------------------------------------------------------------------------------------\n";  cin.get();  goto menu;  case 3:  travqwrep(&front, &rear);  cout << "\n---------------------------------------------------------------------------------------------------------\n\n";  cout << "\tPress any key to return to main menu...";  cout << "\n\n---------------------------------------------------------------------------------------------------------\n";  cin.get();  goto menu;  case 4:  addq(&front, &rear);  cout << "\n---------------------------------------------------------------------------------------------------------\n\n";  cout << "\tPress any key to return to main menu...";  cout << "\n\n---------------------------------------------------------------------------------------------------------\n";  cin.get();  goto menu;  case 5:  delq(&front);  cin.get();  goto menu;  case 6:  return 0;  default:  gotoxy(10, 26);  cout << "WARNING: Invalid Choice!!!\n";  gotoxy(10, 28);  cin.get();  goto menu;  }  }  void MainMenu() {  system("cls");  system("COLOR F1");  cout << "\n========================================== MAIN MENU ====================================================\n";  gotoxy(37, 4);  cout << "Student Record (Queues) ";  gotoxy(36, 5);  cout << "by DSA Group 1 (Searching)\n";  cout << "\n=========================================================================================================\n";  gotoxy(10, 10);  cout << "[1] Create a Student Record\n";  gotoxy(10, 12);  cout << "[2] Traverse Student Record (No Repeating Values)";  gotoxy(10, 14);  cout << "[3] Traverse Student Record (With Repeating Values)";  gotoxy(10, 16);  cout << "[4] Add Student Record\n";  gotoxy(10, 18);  cout << "[5] Delete Student Record\n";  gotoxy(10, 20);  cout << "[6] Exit Program\n";  gotoxy(10, 22);  cout << "\n=========================================================================================================\n";  }  void Create(SRec \*\*front, SRec \*\*rear) {  system("cls");  SRec \*NewRec;  int ctr = 1;  char temp, chc1;  NewRec = new SRec;  NewRec->next = NULL ;  \*front = \*rear = NewRec;  cout << "\n=========================================================================================================\n";  gotoxy(40, 4);  cout << "CREATING A STUDENT RECORD\n";  cout << "\n=========================================================================================================\n";  do {  cout << "\n\tStudent #" << ctr << " Number\t\t: ";  cin >> NewRec->SNum;  cin.ignore();  cout << "\n\tStudent #" << ctr << " Name\t\t\t: ";  cin.getline(NewRec->SName, 25);  cout << "\n\tCourse & Year\t\t\t: ";  cin.getline(NewRec->crsYr, 15);  cout << "\n\tGWA\t\t\t\t: ";  cin >> NewRec->GWA;  cin.ignore();  cout << "\n---------------------------------------------------------------------------------------------------------\n\n";  cout << "\tAdd another record? [Y/N]\t: ";  cin >> chc1;  cin.ignore();  cout << "\n---------------------------------------------------------------------------------------------------------\n\n";  if (toupper(chc1) == 'Y') {  NewRec->next = new SRec;  \*rear = NewRec->next;  NewRec = NewRec->next;  ctr++;  }  } while (toupper(chc1) == 'Y');  (\*rear)->next = NULL ;  NewRec = NULL ;  }  void travqnorep(SRec \*\*front, SRec \*\*rear) {  system("cls");  SRec \*TravNode = \*front;  int ctr = 1;  cout << "\n=========================================================================================================\n";  gotoxy(40, 4);  cout << "STUDENT RECORD\n";  cout << "\n=========================================================================================================\n";  if (TravNode == NULL ) {  gotoxy(35, 8);  cout << "Student Record is empty!\n";  cout << "\n=========================================================================================================\n";  cin.get();  } else {  SRec \*prev = NULL ;  int repeat = 0;  gotoxy(8, 8);  cout << "Student No.";  gotoxy(33, 8);  cout << "Student Name";  gotoxy(63, 8);  cout << "Course & Year";  gotoxy(88, 8);  cout << "GWA\n\n";  do {  repeat = 0;  SRec \*temp = \*front;  while (temp != TravNode) {  if (temp->SNum == TravNode->SNum) {  repeat = 1;  break;  }  temp = temp->next;  }    if (!repeat) {  int i = 10;  while(TravNode != NULL) {  gotoxy(0,i); cout << ctr;  gotoxy(8,i); cout << TravNode->SNum;  gotoxy(33,i); cout << TravNode->SName;  gotoxy(63,i); cout << TravNode->crsYr;  gotoxy(88,i); cout << fixed << setprecision(2) << TravNode->GWA << endl;    i++;  ctr++;  TravNode = TravNode->next;  }  }  prev = TravNode;  } while (TravNode != NULL );  cout << "\n=========================================================================================================\n";  cin.get();  }  }  void travqwrep(SRec \*\*front, SRec \*\*rear) {  system("cls");  SRec \*TravNode = \*front;  int ctr = 1;  cout << "\n=========================================================================================================\n";  cout << "\n";  gotoxy(40, 4);  cout << "STUDENT RECORD\n";  cout << "\n=========================================================================================================\n";  if (TravNode == NULL ) {  gotoxy(35, 8);  cout << "Student Record is empty!\n";  cout << "\n=========================================================================================================\n";  cin.get();  } else {  gotoxy(8, 8); cout << "Student No.";  gotoxy(33, 8); cout << "Student Name";  gotoxy(63, 8); cout << "Course & Year";  gotoxy(88, 8); cout << "GWA\n\n";  int i =10;  do {  gotoxy(0,i); cout << ctr;  gotoxy(8,i); cout << TravNode->SNum;  gotoxy(33,i); cout << TravNode->SName;  gotoxy(63,i); cout << TravNode->crsYr;  gotoxy(88,i); cout << fixed << setprecision(2) << TravNode->GWA << endl;    i++;  ctr++;  TravNode = TravNode->next;  } while (TravNode != NULL );  cout << "\n=========================================================================================================\n";  }  }  void addq(SRec \*\*front, SRec \*\*rear) {  system("cls");  SRec \*NewRec = new SRec;  cout << "\n=========================================================================================================\n";  gotoxy(40, 4);  cout << "ADDING OF STUDENT RECORD IN STACK\n";  cout << "\n=========================================================================================================\n";  if (\*front == NULL ) {  gotoxy(45, 8);  cout << "Student Record is empty!\n";  cout << "\n=========================================================================================================\n";  cin.get();  return;  }  cout << "\n\tStudent Number\t\t\t: ";  cin >> NewRec->SNum;  cin.ignore();  cout << "\n\tStudent Name\t\t\t: ";  cin.getline(NewRec->SName, 25);  cout << "\n\tCourse & Year\t\t\t: ";  cin.getline(NewRec->crsYr, 15);  cout << "\n\tGWA\t\t\t\t: ";  cin >> NewRec->GWA;  if (\*rear == NULL ) {  \*front = \*rear = NewRec;  } else {  (\*rear)->next = NewRec;  \*rear = NewRec;  }  NewRec->next = NULL ;  cout << "\n=========================================================================================================\n";  gotoxy(40, 18);  cout << "Record succesfully added!\n";  cout << "\n=========================================================================================================\n";  cin.get();  }  void delq(SRec \*\*front) {  if (\*front == NULL ) {  gotoxy(10, 26);  cout << "The Queue is empty. Nothing can be deleted.\n";  return;  }  SRec \*temp = \*front;  \*front = (\*front)->next;  delete temp;  gotoxy(10, 26);  cout << "Student record is successfully deleted!\n";  } |
| --- |

1. **Main Menu**

Output

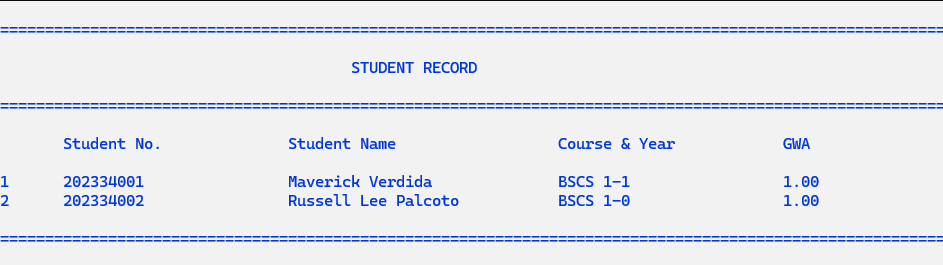
*(Figure 1: Main Menu – Functions)*

1. **Creation of Queues**

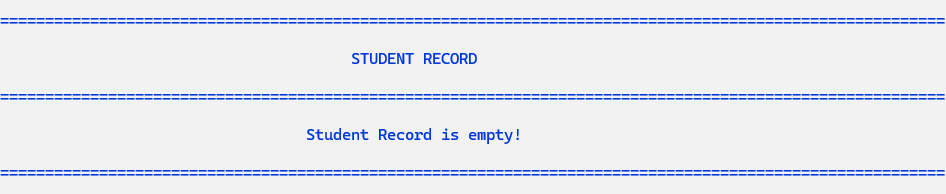
Output

*(Figure 2.1: Entering Data in the “Creating a Student Record” Portion and Selecting “Y” will allow you to input another student record)*

1. **Traversal of Queues (No Repeating Data)**

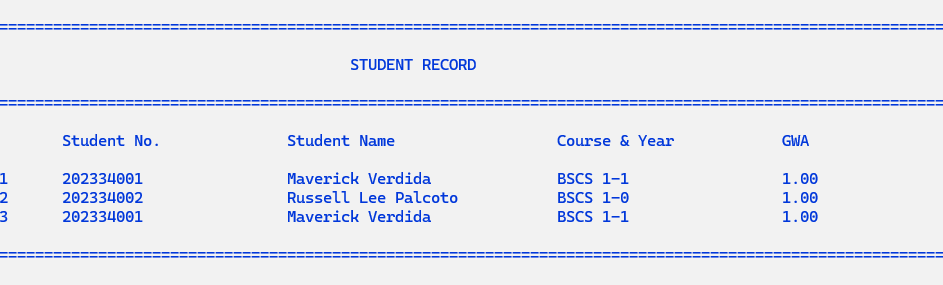
Output

*(Figure 3.1: Displaying the Student Record in Tabulated Format)*

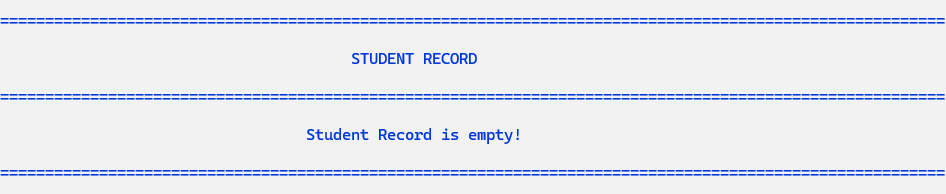
**

*(Figure 3.2: Student Record is Empty)*

1. **Traversal of Queues (With Repeating Data)**

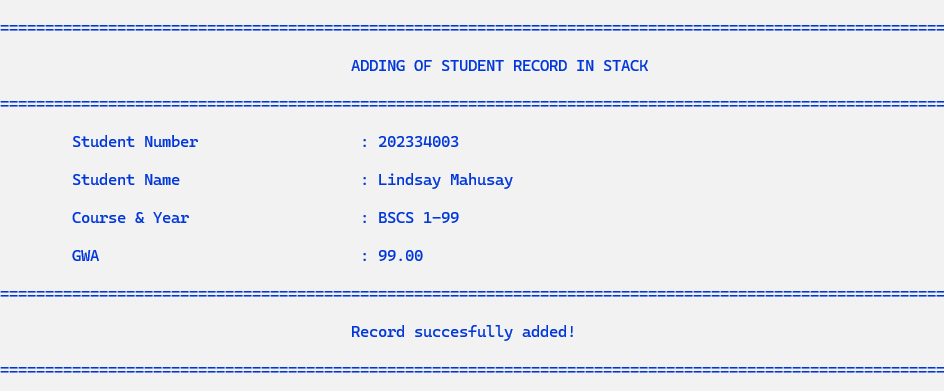
Output

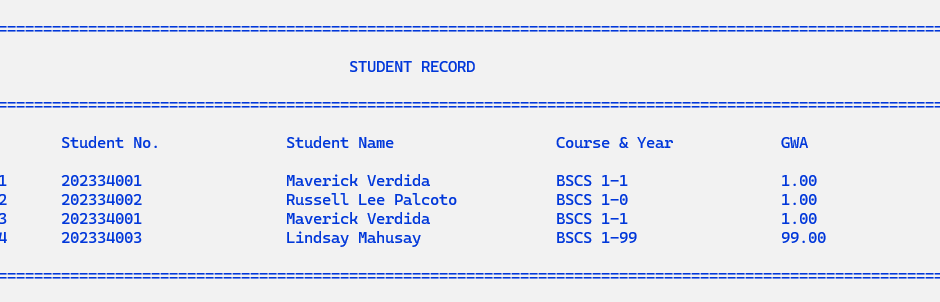
*(Figure 4.1: Displaying the Student Record in Tabulated Format)*



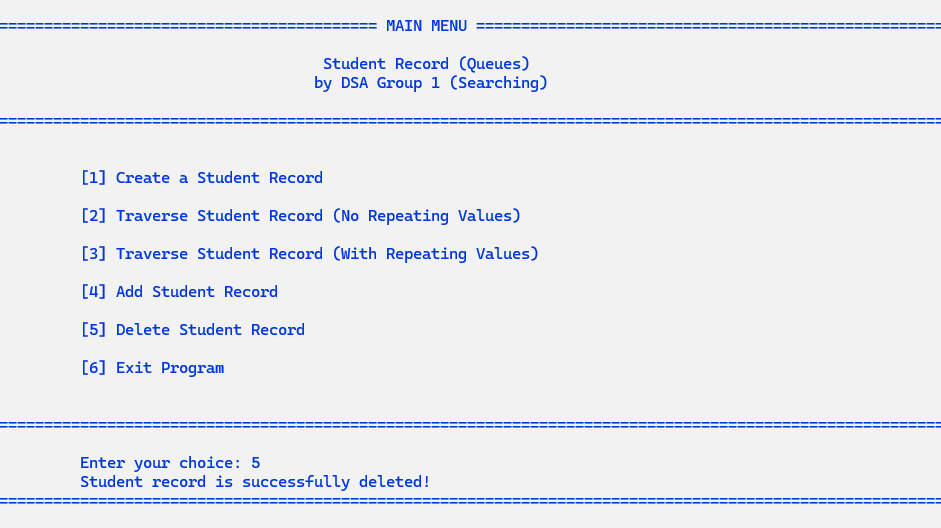
*(Figure 4.2: Student Record is Empty)*

1. **Adding of Node for Queues**

Output

*(Figure 5.1: Adding a new entry to the queue Displaying the Student Record in Tabulated Format)*

1. **Deletion of Node for Queues**

Output

*(Figure 6.1: Deleting an entry to the queues and Displaying the Student Record)*