ព្រះរាជាណាចក្រ​កម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ



វិទ្យាស្ថានបច្ចេកវិទ្យា

ដេប៉ាតឺម៉ង់ថ្នាក់ឆ្នាំសិក្សាមូលដ្ឋាន

កិច្ចការផ្ទះ (I2-GIC-1B)

**មុខវិជ្ជា៖ Data Structure and Programming**

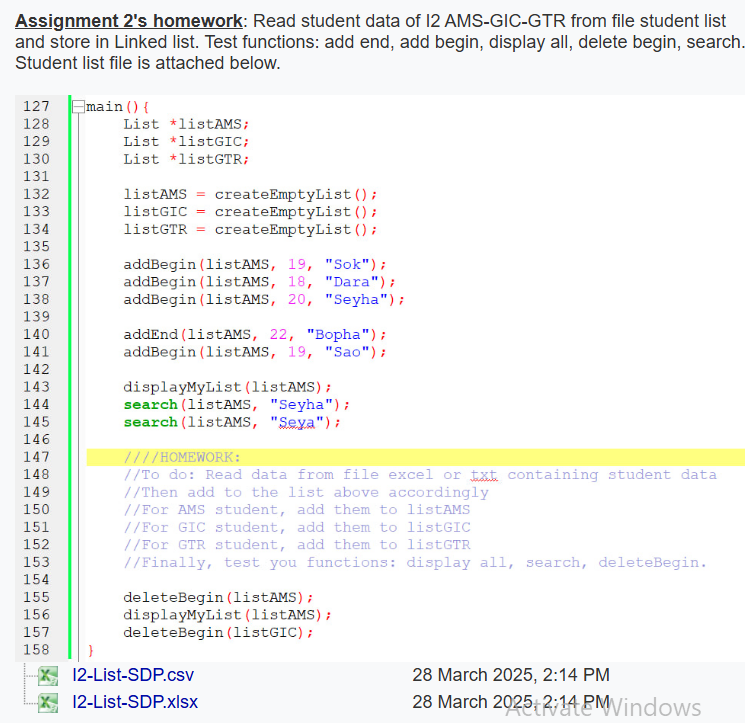
**និស្សិតឈ្មោះ ID ពិន្ទុ**

១. KHORN VICTOR e20230078 ………………………….

បង្រៀនដោយ៖ អ្នកគ្រូ Bou Channa

**ឆ្នាំសិក្សា ២០២៤ - ២០២៥**

**PROBLEM**



**CODE**

#include<iostream>

#include<fstream>

#include<string>

#include<Windows.h>

using *namespace* std;

#define RESET   "\033[0m"

#define RED     "\033[31m"

#define GREEN   "\033[32m"

#define YELLOW  "\033[33m"

#define BLUE    "\033[34m"

#define MAGENTA "\033[35m"

#define CYAN    "\033[36m"

#define WHITE   "\033[37m"

#define BOLD    "\033[1m"

*enum* choice{

    Add\_Product = 1,

    Display\_Product,

    SEaRch,

    DeLEte,

    eXit

};

*struct* Student{

    string id;

    string name;

    string gender;

    string departement;

    Student\* next;

};

*struct* List{

*int* size = 0;

    Student\* head;

    Student\* tail;

};

List *\**create(){

    List \*list = new List;

    list->size = 0;

    list->head = NULL;

    list->tail = NULL;

    return list;

}

*void* add\_back\_to\_file(List *\*list* , string *file\_name*){

    Student \*temporary = list->head;

    ofstream file;

    file.open(file\_name, ios::out);

    while (temporary != NULL){

        if(temporary -> next == NULL) file << temporary -> id << "," << temporary -> name << "," << temporary -> gender << "," << temporary -> departement;

        else file << temporary -> id << "," << temporary -> name << "," << temporary -> gender << "," << temporary -> departement << endl;

        temporary = temporary->next;

    }

    file.close();

}

*void* add\_student\_end\_initial(List *\*list*, string *id*, string *name*, string *gender*, string *departement*){

    Student \*newstudent = new Student;

    newstudent->id = id;

    newstudent->name = name;

    newstudent->gender = gender;

    newstudent->departement = departement;

    newstudent->next = NULL;

    if(list->head == NULL){

        list->head = newstudent;

        list->tail = newstudent;

    }else{

        list->tail->next = newstudent;

        list->tail = newstudent;

    }

    list->size++;

}

*void* add\_student\_end(List *\*list*, string *id*, string *name*, string *gender*, string *departement*){

    Student \*newstudent = new Student;

    newstudent->id = id;

    newstudent->name = name;

    newstudent->gender = gender;

    newstudent->departement = departement;

    newstudent->next = NULL;

    if(list->head == NULL){

        list->head = newstudent;

        list->tail = newstudent;

    }else{

        list->tail->next = newstudent;

        list->tail = newstudent;

    }

    list->size++;

    add\_back\_to\_file(list, "Student.csv");

}

*void* add\_student\_head(List *\*list*, string *id*, string *name*, string *gender*, string *departement*){

    Student \*newstudent = new Student;

    newstudent->id = id;

    newstudent->name = name;

    newstudent->gender = gender;

    newstudent->departement = departement;

    newstudent->next = NULL;

    newstudent->next = list->head;

    list->head = newstudent;

    if (list->head == NULL) list->tail = newstudent;

    list->size++;

    add\_back\_to\_file(list, "Student.csv");

}

*void* display(List *\*list*){

    Student \*temporary = list->head;

*bool* numbered;

*int* i = 1;

*char* check;

    cout << "Do you want to display with number?(Y(es)/N(o))? : "; cin >> check;

    if (check == 'Y' || check == 'y') numbered = true;

    else if (check == 'N' || check == 'n') numbered = false;

    system("cls");

    while(temporary != NULL){

        if (numbered) cout << i << ".\t" << temporary->id << "\t" << temporary -> gender << "\t" << temporary -> departement << "\t" << temporary->name << endl;

        else cout << temporary->id << "\t" << temporary -> gender << "\t" << temporary -> departement << "\t" << temporary->name << "\t" << endl;

        temporary = temporary->next;

        i++;

    }

    cout << endl;

}

*void* get\_data\_from\_file(List *\*list*,string *file\_name*){

    string id, name, gender, departement;

    ifstream file;

*int* i = 0;

    file.open(file\_name);

    if (file.fail()) cout << RED << "File not found" << RESET << "\n" << endl;

    else {

        while(getline(file, id, ',') &&

            getline(file, name, ',') &&

            getline(file, gender, ',') &&

            getline(file, departement, '\n')){

            cout << "Reading: " << id << ", " << name << ", " << gender << ", " << departement << endl;

            i++;

            add\_student\_end\_initial(list, id, name, gender, departement);

        }

    }

    cout << GREEN << "\n" << "Get " << i << " data" << "\n" << RESET << endl;

    file.close();

}

*int* Introduction(){

    cout << "Welcome to the Student Management System" << endl;

    cout << "1. Add Student" << endl;

    cout << "2. Display Student" << endl;

    cout << "3. Search" << endl;

    cout << "4. Delete" << endl;

    cout << "5. Exit" << endl;

    cout << "Choose: ";

*int* choice;

    cin >> choice;

    return choice;

}

*void* Add(List *\*List*){

    Student newstudent;

    cout << "\e[1mAdd Student\e[0m" << endl;

    cout << "Enter Student's ID: ";           cin >> newstudent.id;

    cout << "Enter Student's Name: ";         getline(cin >> ws, newstudent.name);

    cout << "Enter Student's Gender (M/F): "; cin >> newstudent.gender;

    cout << "Enter Student's Departement: ";  getline(cin >> ws, newstudent.departement);

    cout << "\n\n";

*char* choice;

    do{

        cout << "Do you want to add to head or tail? (h/t): ";

        cin >> choice;

        if (choice == 'h') add\_student\_head(List, newstudent.id, newstudent.name, newstudent.gender, newstudent.departement);

        else if (choice == 't') add\_student\_end(List, newstudent.id, newstudent.name, newstudent.gender, newstudent.departement);

        else cout << "Invalid choice\n";

    }while(choice != 'h' && choice != 't');

    system("cls");

    cout << GREEN << "Add Successfully...\n" << RESET << endl;

}

*void* SearchByName(List *\*studentlist*){

    Student \*temporary = new Student; temporary = studentlist -> head;

    string name;cout << "Enter the name of the student you want to search: ";getline(cin >> ws, name);

*bool* found = false;

    system("cls");

    while(temporary != NULL){

        if(temporary -> name == name){

            cout << GREEN << "Student found!" << RESET << "\n" << endl;

            cout << "ID: " << temporary -> id << endl;

            cout << "Name: " << temporary -> name << endl;

            found = true;

            break;

        }

        temporary = temporary -> next;

    }

    if (!found) cout << RED << "Student not found!" << RESET << "\n" << endl;

    cout << "\n";

}

*void* SearchByID(List *\*studentlist*){

    Student \*temporary = new Student; temporary = studentlist -> head;

    string ID;cout << "Enter the id of the student you want to search: ";cin >> ID;

*bool* found = false;

    system("cls");

    while(temporary != NULL){

        if(temporary -> id == ID){

            cout << GREEN << "Student found!" << RESET << "\n" << endl;

            cout << "ID: " << temporary -> id << endl;

            cout << "Name: " << temporary -> name << endl;

            found = true;

            break;

        }

        temporary = temporary -> next;

    }

    if (!found) cout << RED << "Student not found!" << RESET << "\n" << endl;

    cout << "\n";

}

*void* SearchByDepartement(List *\*studentlist*){

    Student \*temporary = new Student; temporary = studentlist -> head;

    string departement;cout << "Enter the departement of the student you want to search: ";getline(cin >> ws, departement);

*bool* found = false;

    system("cls");

    while(temporary != NULL){

        if(temporary -> departement == departement){

            cout << temporary->id << "\t" << temporary -> gender << "\t" << temporary -> departement << "\t" << temporary->name << "\t" << endl;

            found = true;

        }

        temporary = temporary -> next;

    }

    if (!found) cout << "Student not found!" << endl;

    cout << "\n";

}

*void* SearchbyGender(List *\*studentlist*){

    Student \*temporary = new Student; temporary = studentlist -> head;

    string gender;cout << "Enter the gender of the student you want to search: ";cin >> gender;

*bool* found = false;

    system("cls");

    while(temporary!= NULL){

        if(temporary -> gender == gender){

            cout << temporary->id << "\t" << temporary -> gender << "\t" << temporary -> departement << "\t" << temporary->name << "\t" << endl;

            found = true;

        }

        temporary = temporary -> next;

    }

    if (!found) cout << "Student not found!" << endl;

    cout << "\n";

}

*void* Search(List *\*studentList*){

*int* choice;

    do{

        cout << "1. Search by Name" << endl;

        cout << "2. Search by ID" << endl;

        cout << "3. Search by Departement" << endl;

        cout << "4. Search by Gender" << endl;

        cout << "Choose: ";

        cin >> choice;

        switch(choice){

            case 1:SearchByName(studentList);break;

            case 2:SearchByID(studentList);break;

            case 3:SearchByDepartement(studentList);break;

            case 4:SearchbyGender(studentList);break;

            default:cout << "Invalid input\n";continue;

        }

        break;

    }while (choice >= 1 && choice <= 4);

}

void Delete(List \*studentlist){

    Student \*temporary = new Student;

    Student \*previous = new Student;

    string ID;cout << "Enter the ID of the student you want to delete: ";cin >> ID;

    temporary = studentlist -> head;

    bool found = false;

    while(temporary != NULL){

        if(temporary -> id== ID){

            if(temporary == studentlist -> head) studentlist -> head = studentlist -> head -> next;

            else if(temporary == studentlist -> tail){

                studentlist -> tail = previous;

                previous -> next = NULL;

            }

            else previous -> next = temporary -> next;

            cout << GREEN << "Student deleted!" << RESET << endl;

            found = true;

            break;

        }

        previous = temporary;

        temporary = temporary -> next;

    }

    if (!found) cout << RED << "Student not found!" << RESET << endl;

    add\_back\_to\_file(studentlist, "Student.csv");

    cout << "\n";

}

void Exit(int number, List \*Student\_list){

    for (int i = 0; i < 3; i++){

        system("cls");

        if (i >= 2) cout << "Exiting in " << 3 - i << " second..." << endl;

        else cout << "Exiting in " << 3 - i << " seconds..." << endl;

        Sleep(1000);

    }

    system("cls");

    cout << GREEN << "Program terminated successfully" << RESET << endl;

    cout << "Thank you for using the student Management System" << endl;

    cout << "Goodbye!" << endl;

    add\_back\_to\_file(Student\_list ,"Student.csv");

    exit(0);

}

int main(){

    ifstream file;

    List \*Student\_list = create();

    cout << GREEN << "Program start...\n" << RESET << endl;

    get\_data\_from\_file(Student\_list ,"Student.csv");

    while (true){

        switch (Introduction()){

        case Add\_Product:system("cls");Add(Student\_list);break;

        case Display\_Product:system("cls");display(Student\_list);break;

        case SEaRch:system("cls");Search(Student\_list);break;

        case DeLEte:system("cls");Delete(Student\_list);break;

        case eXit:system("cls");Exit(0, Student\_list);break;

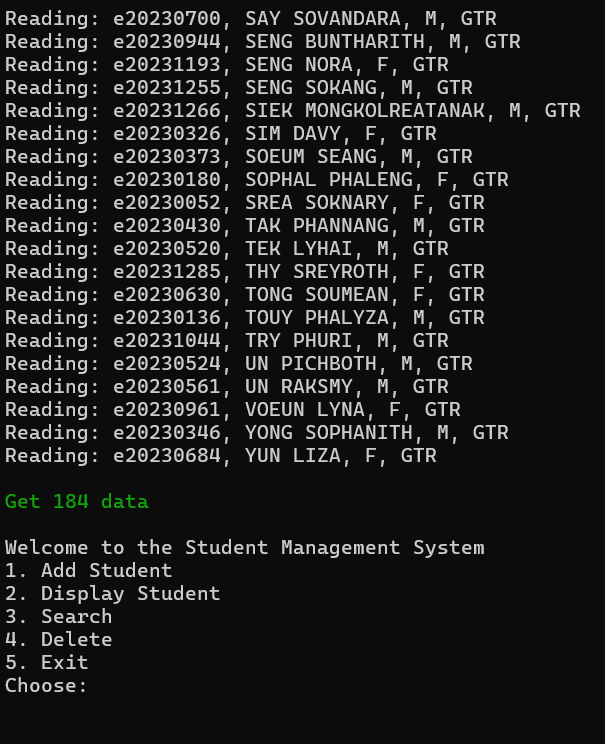
        default:cout << "Invlid intput\n";break;

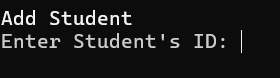
        }

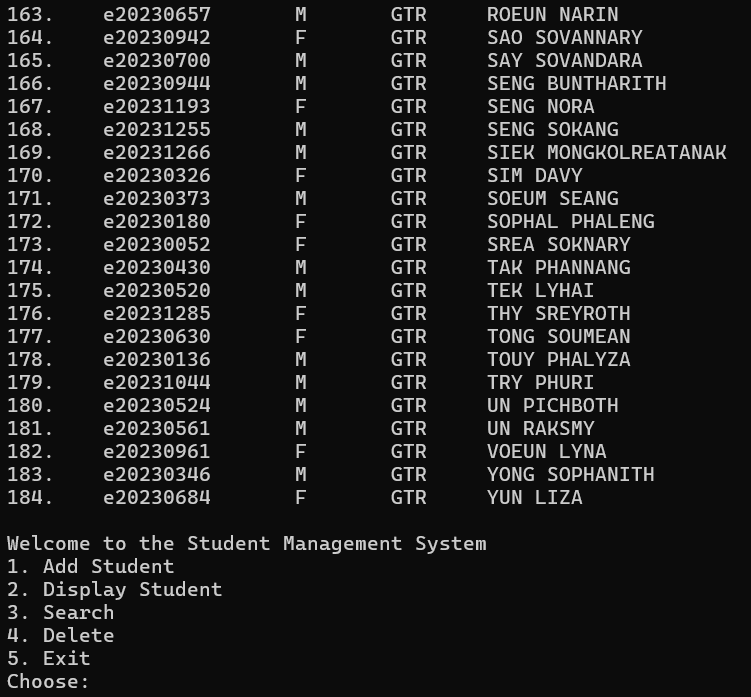
    }

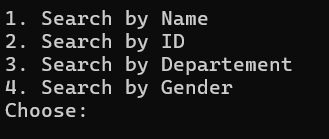
}

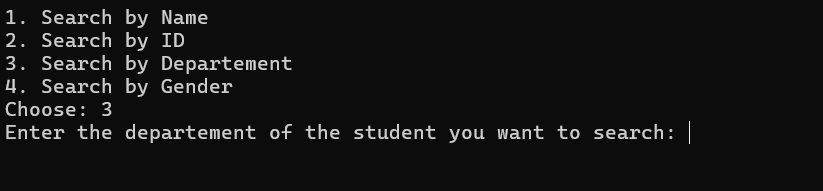
**RESULT**

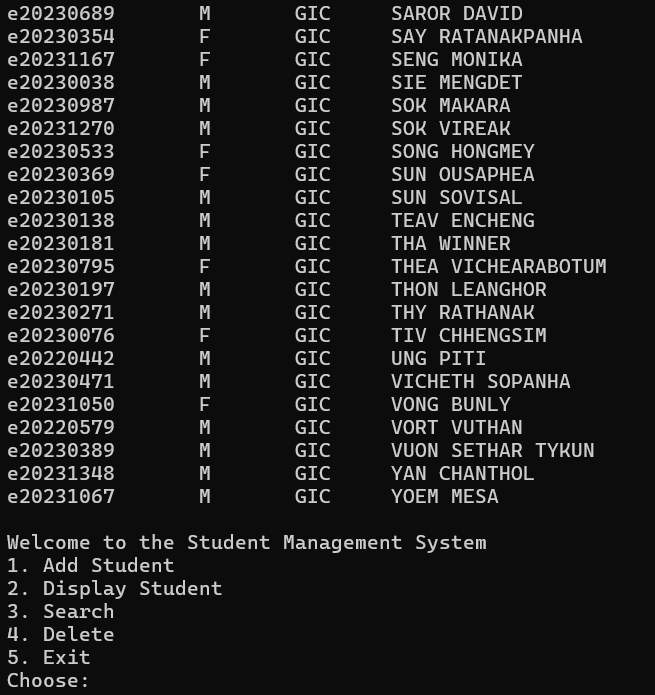
****

****

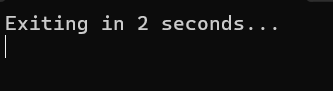
****

****

****

****

****

****