Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

**Кафедра информационных технологий**

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**(практика по получению первичных профессиональных умений и навыков)**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П.Редько

студент ФКТиПМ, 1 курса, 11 гр.,

направление 01.03.02 «Прикладная математика и информатика»

Руководитель учебной практики

(практика по получению

первичных профессиональных

умений и навыков) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П.Лукащик

к.ф.-м.н., доцент кафедры

информационных технологий

Краснодар 2019

1. ***Условие задачи:***

Каждая запись вводного файла содержит результат одного из участников универсиады в следующем формате

<Фамилия> <Имя> <курс> <баллы>

где <Фамилия> – строка, состоящая не более, чем из 30 символов, <Имя> – строка, состоящая не более, чем из 15 символов, <курс> – число от 1 до 4, <баллы> – целое число от 0 до 70 набранных участником баллов.

Для определения призеров универсиады сначала отбираются 25% участников, показавших лучшие результаты. Если у последнего участника, входящего в 25%, оказывается такое же количество баллов как и у следующих за ним в итоговой таблице, все они считаются призерами только тогда, когда набранные ими баллы больше половины максимально возможных; иначе все они не считаются призерами.

Напишите программу, которая по результатам универсиады будет определять минимальный балл призера универсиады, и количество призеров было в каждой параллели (среди 1-х, 2-х, 3-х, и 4-х курсов отдельно).

1. ***Описание алгоритма решения и структур хранения данных:***

Осуществим запись данных в однонаправленный список student , состоящий из поля part , представленного структурой Participant, состоящий из полей surname(фамилия участника) и name ( имя участника , представленных массивом типа char, и полей kurs и points типа int , поля next на следующий элемент списка.

Функция **Fill\_input\_file** заполняет бинарный файл **input.bin** значениями, введёнными с клавиатуры в консоль (Фамилию и имя участника универсиады, его курс и заработанные баллы

Функция **Add\_student** добавляет элемент (студента с фамилией, именем, курсом и баллами) в список студентов следующим образом: студенты записываются в список в порядке убывания баллов, в списке ограниченное количество мест (25% от количества всех участников). Если список полон и добавляется еще один элемент, который больше минимального, то минимальный элемент удаляется (и все его вхождения в списке), если без учета всех минимальных элементов количество элементов в списке равняется 25% от всех участников, и не удаляются, если хотя бы 1 минимальный элемент содержится в 25%.

Функция **Fill\_list** заполняет список **lst** данными из файла **input.bin** используя функцию Add\_student.

Функция **Count** считает количество призеров для каждого курса , а также подсчитывает минимальный балл призеров универсиады.

Таким образом используются следующие переменные и типы:

1. **input.bin** содержит входные данные, введенные через консоль
2. Структура **Participant**с полями:
   1. Массив **surname**(*char*), состоящий из 30 элементов, содержит фамилию участника.
   2. Массив name(char), состоящий из 15 элементов, содержит имя участника
   3. **kurs** (*int*) содержит курс студента
   4. **points** (*int*) содержит количество заработанных баллов
3. Структура **list** с полями:
   1. Структура **part** (*Participant*)
   2. Указатель на следующий элемент в списке next.
4. Структура AllStud содержит поля kol1, kol2, kol3, kol4 содержащие количество призеров среди первых, вторых, третьих, четвертых курсов соответственно.
5. **InName** (*const char*) содержит указатель на начало входного файла.
6. **n** (*int*) содержит количество участников универсиады
7. Список **lst** (*list*) содержит данные из входного файла
8. Структура point(AllStud) содержит количество призеров в каждом курсе.
9. ***Описание входных и выходных данных:***

Каждая запись вводного файла содержит результат одного из участников универсиады в следующем формате

**<Фамилия> <Имя> <курс> <баллы>**

где <Фамилия> – строка, состоящая не более, чем из 30 символов, <Имя> – строка, состоящая не более, чем из 15 символов, <курс> – число от 1 до 4, <баллы> – целое число от 0 до 70 набранных участником баллов.

Напишите программу, которая по результатам универсиады будет определять минимальный балл призера универсиады, и количество призеров было в каждой параллели (среди 1-х, 2-х, 3-х, и 4-х курсов отдельно).

1. ***Текст программы:***

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

using namespace std;

struct Participant

{

char surname[30];

char name[15];

int kurs;

int points;

};

struct list

{

Participant part;

list \*next;

};

struct AllStud

{

int kol1;

int kol2;

int kol3;

int kol4;

};

void Fill\_input\_file(int n, const char \*FName)

{

Participant dop;

ofstream file(FName, ios::binary);

for (int i = 0; i < n; i++)

{

cout << " студент " <<i+1<< endl;

cin >> dop.surname;

cin >> dop.name;

cin >> dop.kurs;

cin >> dop.points;

file.write((char\*)&dop, sizeof(Participant));

}

file.close();

}

void Add\_students(list \*lst,char \*surname, char \*name, int kurs,int points,int &k,int n,int &min)

{

list \*tmp=lst;

if (k < n)

{

while (tmp->next != 0 && tmp->next->part.points >= points)

tmp = tmp->next;

list \*r = new list;

strncpy(r->part.surname, surname, 30);

strncpy(r->part.name, name, 15);

r->part.kurs = kurs;

r->part.points = points;

r->next = tmp->next;

tmp->next = r;

if (points < min) min = points;

k++;

}

else if (points == min)

{

while (tmp->next != 0) tmp = tmp->next;

list \*r = new list;

strncpy(r->part.surname, surname, 30);

strncpy(r->part.name, name, 15);

r->part.kurs = kurs;

r->part.points = points;

r->next = tmp->next;

tmp->next = r;

k++;

min = points;

}

else

{

int kol = 0;

while (tmp->next != 0 && tmp->next->part.points >= points)

{

tmp = tmp->next;

kol++;

}

if (kol < k)

{

list \*r = new list;

strncpy(r->part.surname, surname, 30);

strncpy(r->part.name, name, 15);

r->part.kurs = kurs;

r->part.points = points;

r->next = tmp->next;

tmp->next = r;

k++;

while (tmp->next->part.points != min)

{

tmp = tmp->next;

kol++;

}

if (kol+1 > n)

{

tmp->next = 0;

k = kol;

}

min = tmp->part.points;

}

}

}

void Fill\_list(list \*lst, const char \*FName,int n,int &min)

{

Participant dopPart;

int k = 0;

min = 70;

ifstream file(FName, ios::binary | ios::in);

while (!file.eof())

{

file.read((char\*)&dopPart, sizeof(dopPart));

Add\_students(lst,dopPart.surname,dopPart.name, dopPart.kurs, dopPart.points,k,n,min);

}

file.close();

lst = lst->next;

}

AllStud Count(list \*lst,int min)

{

int minim = 70;

AllStud all = {0,0,0,0 };

while (lst)

{

if (lst->part.points == min && lst->part.points <= 35) break;

switch (lst->part.kurs)

{

case 1:

all.kol1++;

break;

case 2:

all.kol2++;

break;

case 3:

all.kol3++;

break;

case 4:

all.kol4++;

break;

}

minim = lst->part.points;

lst = lst->next;

}

cout<<"Минимальный балл призера универсиады = "<<minim<<endl;

return all;

}

int main()

{

setlocale(LC\_ALL, "rus");

list \*students = new list;

int min;

strncpy(students->part.surname, "Example", 30);

strncpy(students->part.name, "Example", 15);

students->part.kurs = 0;

students->part.points = 70;

students->next = 0;

AllStud point = { 0,0,0,0 };

const char \*InName("input.bin");

int n;

cout << "Введите количество людей" << endl;

cin >> n;

Fill\_input\_file(n, InName);

Fill\_list(students, InName, n / 4, min);

if (n / 4 != 0) point = Count(students,min);

if (point.kol1) cout << "Количество призеров первого курса = " << point.kol1 << endl;

else cout<< "Нет призеров первого курса" << endl;

if (point.kol2) cout << "Количество призеров второго курса = " << point.kol2 << endl;

else cout << "Нет призеров второго курса" << endl;

if (point.kol3) cout << "Количество призеров третьего курса = " << point.kol3 << endl;

else cout << "Нет призеров третьего курса" << endl;

if (point.kol4) cout << "Количество призеров четвертого курса = " << point.kol4 << endl;

else cout << "Нет призеров четвертого курса" << endl;

system("pause");

return 0;

}

1. ***Тестовый запуск программы и вывод результатов работы:***

В входном файле(**input.bin**) содержится информация о 15-ти участниках универсиады:

Голубятников Андрей 1 60

Супрун Дарья 2 65

Джерипа Александр 1 58

Алексеев Илья 3 60

Магомедова Зарема 1 30

Давыденко Андрей 2 35

Юматов Глеб 3 20

Аймальдинова Вера 4 59

Багаевская Екатерина 1 60

Диденко Мария 4 60

Косяков Дмитрий 2 38

Носков Александр 1 59

Приходько Андрей 3 69

Штуркина Анна 4 25

Наумович Юлия 3 58

Результатом является минимальный балл призера универсиады и количество призеров в каждой параллели:

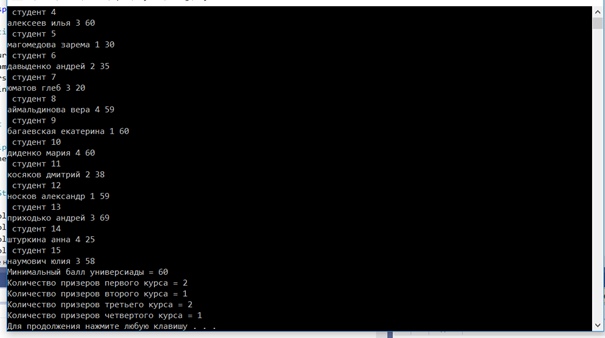
Минимальный балл призера универсиады = 60

Количество призеров первого курса = 2

Количество призеров второго курса = 1

Количество призеров третьего курса = 2

Количество призеров четвертого курса = 1



Программа работает корректно.