**Subject: PRF192- PFC**

**Workshop 06**

**Objectives: Managing arrays**

**Tên: Trần Nguyễn Quốc Cường**

Tiếp thu 90%, điểm slot: 8đ

**Problem 1 (4 marks)**

/\*

Name: Tran Nguyen Quoc Cuong

Date: 2022/03/02

\*/

#include <stdio.h>

int isbn[10];

void arrayConvert(int input) {

for (int i = 9; i > 0; i--) {

isbn[i] = input % 10;

input /= 10;

}

}

void checkCanadianSIN() {

int sum = 0;

int counter = 10;

for (int i = 0; i < 9; i++) {

sum += isbn[i] \* counter;

counter--;

}

sum += isbn[9];

if (sum % 11 == 0) {

printf("This is a valid ISBN\n");

}

else {

printf("This is not a valid ISBN\n");

}

}

int main() {

int input;

printf("ISBN Validator ==============\n");

printf("\nISBN (0 to quit): ");

scanf("%d", &input);

if (input == 0) {

printf("\nHave a Nice Day!");

}

else {

arrayConvert(input);

checkCanadianSIN();

}

return 0;

}

**Problem 2 (6 marks)**

/\*

Name: Tran Nguyen Quoc Cuong

Date: 2022/03/02

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

double\* arr;

short currIdx = 0;

short counter = 2;

void push(double value) {

arr[currIdx] = value;

arr = (double\*)realloc(arr, (size\_t)counter \* sizeof(double));

++currIdx;

++counter;

}

void printAllSearchValueExistences(double searchValue) {

short countSearchValue = 0;

printf("\nSo luong: ");

for (int idx = 0; idx < currIdx; ++idx) {

if (arr[idx] == searchValue) {

++countSearchValue;

}

}

printf("%hi\n", countSearchValue);

}

void printArray() {

printf("\nArray: ");

for (int idx = 0; idx < currIdx; ++idx) {

printf("%lf ", arr[idx]);

}

printf("\n");

}

void printArrayInRange(double minValue, double maxValue) {

printf("\nArray between min and max: ");

for (int idx = 0; idx < currIdx; ++idx) {

if (arr[idx] >= minValue && arr[idx] <= maxValue) {

printf("%lf ", arr[idx]);

}

}

printf("\n");

}

int inc(const void\* a, const void\* b) {

if (\*(double\*)a > \*(double\*)b) {

return 1;

}

return -1;

}

void sortedArrayPrint() {

double\* tmpArr = (double\*)malloc(currIdx \* sizeof(double));

memcpy(tmpArr, arr, currIdx \* sizeof(double));

qsort(tmpArr, currIdx, sizeof(double), inc);

printf("\nArray after sorted: ");

for (int idx = 0; idx < currIdx; ++idx) {

printf("%lf ", tmpArr[idx]);

}

printf("\n");

}

void userMenuPrinting() {

printf("\n--- MENU ---\n");

printf("\n1- Add a value\n");

printf("2- Search a value\n");

printf("3- Print out the array\n");

printf("4- Print out values in a range\n");

printf("5- Print out the array in ascending order\n");

printf("Others- Quit\n");

}

void getCheckUserInput() {

int input;

do {

printf("\nNhap lua chon: ");

scanf("%d", &input);

switch (input) {

case 1:

{

double value;

printf("\nNhap gia tri can them: ");

scanf("%lf", &value);

push(value);

break;

}

case 2:

{

double searchValue;

printf("\nNhap gia tri can tim: ");

scanf("%lf", &searchValue);

printAllSearchValueExistences(searchValue);

break;

}

case 3:

{

printArray();

break;

}

case 4:

{

double minVal;

double maxVal;

printf("\nNhap Min: ");

scanf("%lf", &minVal);

printf("\nNhap Max: ");

scanf("%lf", &maxVal);

printArrayInRange(minVal, maxVal);

break;

}

case 5:

{

sortedArrayPrint();

break;

}

}

} while ((input > 0) && (input < 6));

}

int main() {

arr = (double\*)calloc(1, sizeof(double));

userMenuPrinting();

getCheckUserInput();

return 0;

}