**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**



**LAB REPORT EXERCISES**

**KỸ THUẬT LẬP TRÌNH**

## MSSV: 23727341 Họ Tên: Mai Văn Khai

**NỘI DUNG 1: CON TRỎ**

**Câu 2:** Dùng con trỏ, viết chương trình để thực hiện các phép tính số học.

**Code:**

#include <stdio.h>

int main() {

int a = 9;

int b = 5;

int \*ptr1, \*ptr2;

ptr1 = &a;

ptr2 = &b;

//Phép tính cộng

int sum = \*ptr1 + \*ptr2;

printf("Gia tri phep tinh cong su dung con tro la: %d\n", sum);

//Phép tính trừ

int sub = \*ptr1 - \*ptr2;

printf("Gia tri phep tinh tru su dung con tro la: %d\n", sub);

//Phép tính nhân

int mult = \*ptr1 \* \*ptr2;

printf("Gia tri phep tinh nhan su dung con tro la: %d\n", mult);

//Phép tính chia

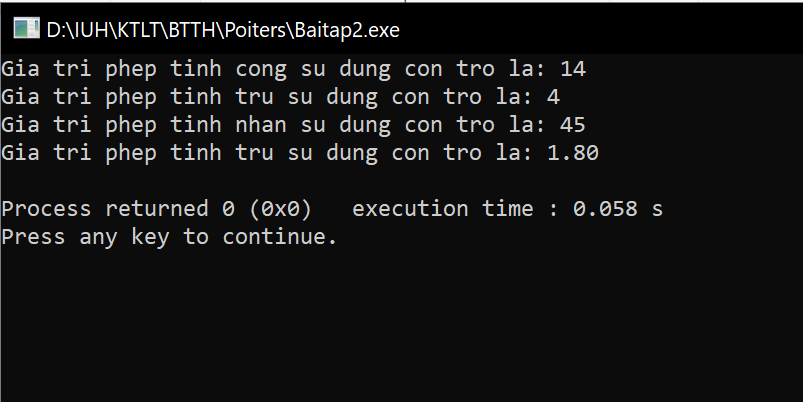
float div = (float)\*ptr1 / (float)\*ptr2;

printf("Gia tri phep tinh chia su dung con tro la: %.2f\n", div);

return 0;

}

**Kết quả:**



**Câu 3:** Dùng con trỏ, viết chương trình để tìm giá trị lớn nhất giữa hai số.

**Code:**

#include <stdio.h>

int getMax(int \*ptr1, int \*ptr2);

int main() {

int a = 100;

int b = -5;

int \*ptr1, \*ptr2;

ptr1 = &a;

ptr2 = &b;

int max\_val = getMax(ptr1, ptr2);

if(max\_val == 0)

printf("Khong co gia tri lon nhat\n");

else

printf("Gia tri lon nhat giua hai so %d va %d la: %d\n", \*ptr1, \*ptr2, max\_val);

return 0;

}

int getMax(int \*ptr1, int \*ptr2) {

int result;

if (\*ptr1 > \*ptr2)

result = \*ptr1;

else if (\*ptr1 < \*ptr2)

result = \*ptr2;

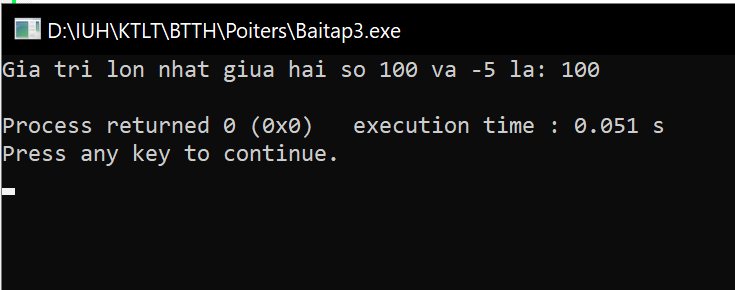
else

result = 0;

return result;

}

**Kết quả:**



**Câu 4:** Dùng con trỏ, viết hàm hoán đổi hai số (swap).

**Code:**

#include <stdio.h>

void swap(int \*x, int \*y);

int main() {

int a = 36;

int b = 45;

int \*ptr1, \*ptr2;

ptr1 = &a;

ptr2 = &b;

printf("Gia tri cua a truoc khi hoan doi la: %d\n", \*ptr1);

printf("Gia tri cua b truoc khi hoan doi la: %d\n", \*ptr2);

swap(ptr1, ptr2);

printf("Gia tri cua a sau khi hoan doi la: %d\n", \*ptr1);

printf("Gia tri cua b sau khi hoan doi la: %d\n", \*ptr2);

return 0;

}

void swap( int \*x, int \*y) {

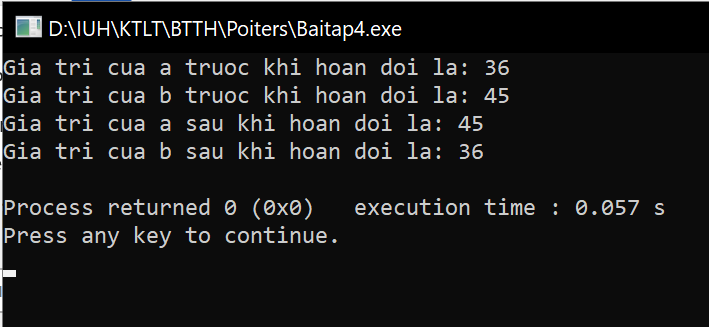
int m = \*x;

\*x = \*y;

\*y = m;

}

**Kết quả:**



**Câu 5:** Viết chương trình thực hiện các chức năng sau:

a) Nhập mảng số nguyên.

b) Xuất ra mảng số nguyên vừa nhập.

c) Xuất các số chẵn có trong mảng.

d) Tìm giá trị lớn nhất trong mảng.

**Code:**

#include <stdio.h>

#define MAXN 100

void inputArr(int \*a, int \*pn);

void printArr(int \*a, int n);

void printEven(int \*a, int n);

void printMax(int \*a, int n);

int main() {

int a[MAXN];

int n;

int maxVal;

// a) Nhập mảng số nguyên.

inputArr(a, &n);

//b) Xuất ra mảng số nguyên vừa nhập

printArr(a, n);

//c) Xuất các số chẵn có trong mảng

printEven(a, n);

// d) Tìm giá trị lớn nhất trong mảng.

printMax(a, n);

return 0;

}

void inputArr(int \*a, int \*pn) {

\*pn = 0;

printf("Nhap toi da %d phan tu, 0 de dung nhap: ", MAXN);

int x;

do {

scanf("%d", &x);

if(x != 0)

a[(\*pn)++] = x; // Su dung hau to (postfix) ++ đe tra ve bien coppy cua pn dau tien la 0

if(x == 0 && \*pn == 0)

printf("Vui long nhap it nhat 1 phan tu vao mang: ");

} while((x!=0 && \*pn < MAXN) || \*pn == 0);

}

void printArr(int \*a, int n) {

printf("Cac phan tu co trong mang la: ");

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

printf("%d ", \*(a + i));

}

printf("\n");

}

void printEven (int \*a, int n) {

printf("Cac so chan trong mang la: ");

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

if(\*(a + i) % 2 == 0)

printf("%d ", \*(a + i));

}

printf("\n");

}

void printMax(int \*a, int n) {

printf("Gia tri lon nhat co trong mang la: ");

if(n==1) {

printf("%d", \*a);

return;

}

int max\_val = \*a;

for(int i = 1; i < n; i++){

if( max\_val < \*(a + i))

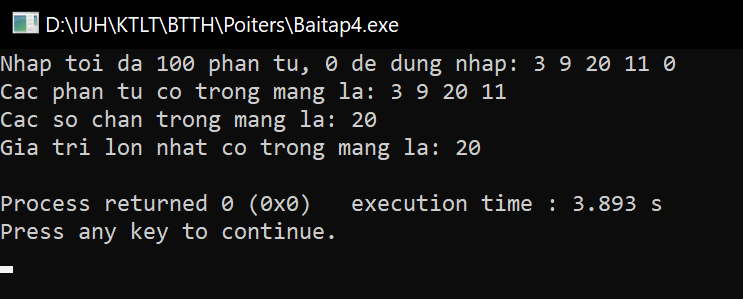
max\_val = \*(a + i);

}

printf("%d\n", max\_val);

}

**Kết quả:**

****

**Câu 6:** Viết chương trình sử dụng vùng bộ nhớ cấp phát động (HEAP SEGMENT) cho phép người dùng nhập 2 số thực. Tính tổng, hiệu, tích, thương của chúng.

**Code:**

#include <stdio.h>

#include <stdlib.h>

int main() {

float \*ptr1, \*ptr2;

ptr1 = (float\*)malloc(sizeof(float));

ptr2 = (float\*)malloc(sizeof(float));

printf("Nhap 2 so thuc: ");scanf("%f%f", ptr1, ptr2);

//Tổng

float sum = \*ptr1 + \*ptr2;

printf("Tong cua %f + %f = %f\n", \*ptr1, \*ptr2, sum);

//Hiệu

float sub = \*ptr1 - \*ptr2;

printf("Hieu cua %f - %f = %f\n", \*ptr1, \*ptr2, sub);

//Tích

float mult = \*ptr1 \* \*ptr2;

printf("Tich cua %f \* %f = %f\n", \*ptr1, \*ptr2, mult);

//Thương

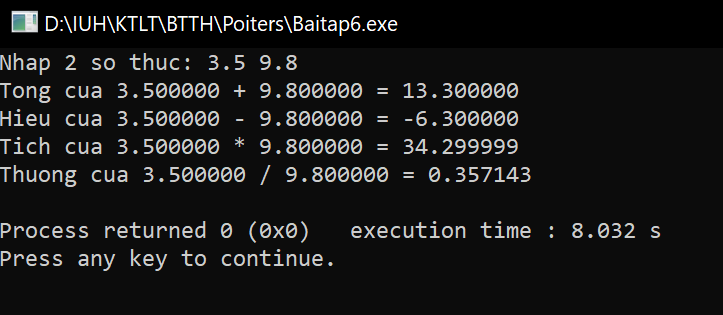
float div = (float)\*ptr1 / (float)\*ptr2;

printf("Thuong cua %f / %f = %f\n", \*ptr1, \*ptr2, div);

return 0;

}

**Kết quả:**



**Câu 7:** Viết chương trình sử dụng bộ nhớ cấp phát động cho phép người dùng nhập 2 ký tự, sau đó chương trình sẽ in ra các ký tự giữa 2 ký tự này theo thứ tự tăng dần.

**Code:**

#include <stdio.h>

#include <stdlib.h>

void swap(char \*x, char \*y);

int main() {

char \*pc1, \*pc2;

pc1 = (char\*)malloc(sizeof(char));

pc2 = (char\*)malloc(sizeof(char));

printf("Nhap 2 ky tu: ");scanf("%c%c", pc1, pc2);

// Hoan vi neu chu truoc lon hon chu sau

if(\*pc1 > \*pc2)

swap(pc1, pc2);

// Hien thi ky tu tang dan

for(int i = \*pc1; i <= \*pc2; i++) {

printf("\n%c %4d %4o %4X", i, i, i, i);

}

return 0;

}

void swap( char \*x, char \*y) {

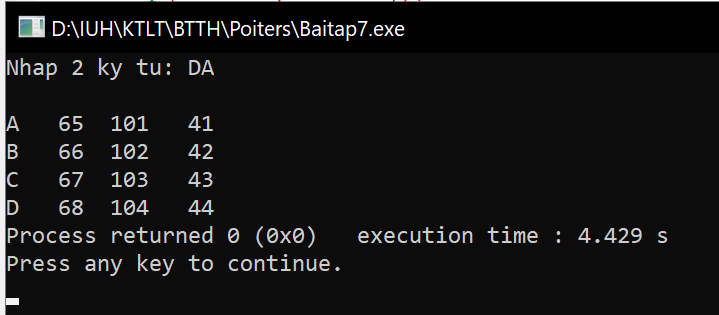
int m = \*x;

\*x = \*y;

\*y = m;

}

**Kết quả:**

****

**Câu 8:** Viết chương trình thực hiện các thao tác trên mảng 1 chiều, sử sụng kỹ thuật cấp phát động (HEAP).

a) Nhập mảng số thực.

b) Xuất các phần tử của mảng.

c) Tính giá trị trung bình của mảng: m = ( x0 + x1 + x2 + ... + xn-1 ) / n

d) Tính tổng bình phương: ss = x0 2 + x1 2 + x2 2 + ... + xn-1 2

e) Tính phương sai (Variance): d 2 = ( ss / n ) - m2

f) Tính độ lệch chuẩn (Standard deviation): d = sqrt( ( ss / n ) - m2 ).

g) Thêm giá trị X vào vị trí P (X, P nhập từ bàn phím)

h) Xóa phần tử tại ví trí P (P nhập từ bàn phím)

i) Tìm kiếm giá trị X trong mảng (X nhập từ bàn phím)

j) Sắp xếp mảng tăng/giảm

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define MAXN 100

void inputArr(float \*a, int \*n);

void printArr(float \*a, int n);

double average(float \*a, int n);

double sumOfSquares(float \*a, int n);

double variance(double ss, double m, int n);

double standardDeviation(double dd);

void insertElement(float \*a, float x, int p, int \*n);

void del(float \*a, int \*pt, int p);

int search(float \*a, int \*pt, float x);

void sortAscending(float \*a, int \*pt);

void sortDescending(float \*a, int \*pt);

int main() {

float a[MAXN], x;

int n, p;

int \*pt = &n;

// a) Nhap mang so thuc

inputArr(a, pt);

// b) Xuat cac phan tu mang

printArr(a, n);

// c) Tính giá trị trung bình của mảng

double m = average(a, n);

printf("Gia tri trung binh cua mang la: %lf\n", m);

// d) Tính tổng bình phương

double ss = 0.0;

ss = sumOfSquares(a, n);

printf("Tong binh phuong la: %lf\n", ss);

// e) Tính phương sai

double dd = variance(ss, m, n);

printf("Phuong sai la: %lf\n", dd);

// e) Tính độ lệch chuẩn

double d = standardDeviation(dd);

printf("Do lech chuan la: %lf\n", d);

// g) Thêm giá trị X vào vị trí P (X, P nhập từ bàn phím);

printf("\nNhap gia tri can chen x: "); scanf("%f", &x);

do {

printf("\nNhap vi tri can chen p: "); scanf("%d", &p);

} while(p > (MAXN-1) || p < 0);

insertElement(a, x, p, pt);

printArr(a, n);

// h) Xóa phần tử tại ví trí P (P nhập từ bàn phím)

do {

printf("\nNhap vi tri can xoa p: "); scanf("%d", &p);

} while(p > (\*pt -1) || p < 0);

del(a, pt, p);

printArr(a, n);

// i) Tìm kiếm giá trị X trong mảng (X nhập từ bàn phím)

printf("\nNhap so can tim x: ");scanf("%f", &x);

int kq = search(a, pt, x);

if(kq==-1) printf("Khong tim thay %f trong mang\n", x);

else printf("Tim thay %f trong mang o vi tri %d\n", x, kq);

// j) Sắp xếp mảng tăng dần

sortAscending(a, pt);

printf("Sap xep mang tang dan\n");

printArr(a, n);

// j) Sắp xếp mảng giam dần

sortDescending(a, pt);

printf("Sap xep mang giam dan\n");

printArr(a, n);

}

void inputArr(float \*a, int \*pn) {

\*pn = 0;

printf("Nhap mang so thuc, toi da %d phan tu, 0 de dung nhap: ", MAXN);

float x;

do {

scanf("%f", &x);

if(x != 0.00)

a[(\*pn)++] = x; // Su dung hau to (postfix) ++ đe tra ve bien coppy cua pn dau tien la 0

if(x == 0 && \*pn == 0)

printf("Vui long nhap it nhat 1 phan tu vao mang: ");

} while((x!=0.00 && \*pn < MAXN) || \*pn == 0);

}

void printArr(float \*a, int n) {

printf("Cac phan tu co trong mang la: ");

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

printf("%f ", \*(a + i));

}

printf("\n");

}

double average(float \*a, int n) {

double m = 0.0;

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

m += \*(a + i);

}

return (m /n);

}

double sumOfSquares(float \*a, int n) {

double ss = 0.0;

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

ss += pow(\*(a + i), 2);

}

return ss;

}

double variance(double ss, double m, int n) {

double d = (ss/n) - pow(m, 2);

return pow(d, 2);

}

double standardDeviation(double dd) {

return sqrt(dd);

}

void insertElement(float \*a, float x, int p, int \*pt) {

if ( p >= \*pt ) {

a[\*pt] = x;

} else {

for(int i=\*pt; i > p; i--) a[i]=a[i-1];

a[p]=x;

}

(\*pt)++;

}

void del(float \*a, int \*pt, int p) {

if(\*pt==0)printf("\nEmpty!");

else{

for(int i=p; i<\*pt; i++)a[i]=a[i+1];

(\*pt)--;

}

}

int search(float \*a, int \*pt, float x) { // a={1,5,3,7}; n=4; x=9

int i = 0;

while(i<\*pt && a[i]!=x)i++;

if(i==\*pt)return -1; else return i;

}

void sortAscending(float \*a, int \*pt) {

for(int i=0; i<\*pt; i++) {

for(int j=i+1; j<\*pt; j++){ //9 2 3 8 5 9

if(a[i]>a[j]){

a[i] = a[i]+a[j];

a[j] = a[i]-a[j];

a[i] = a[i]-a[j];

}

}

}

}

void sortDescending(float \*a, int \*pt) {

for(int i=0; i<\*pt; i++) {

for(int j=i+1; j<\*pt; j++){

if(a[i]<a[j]){

a[i] = a[i]+a[j];

a[j] = a[i]-a[j];

a[i] = a[i]-a[j];

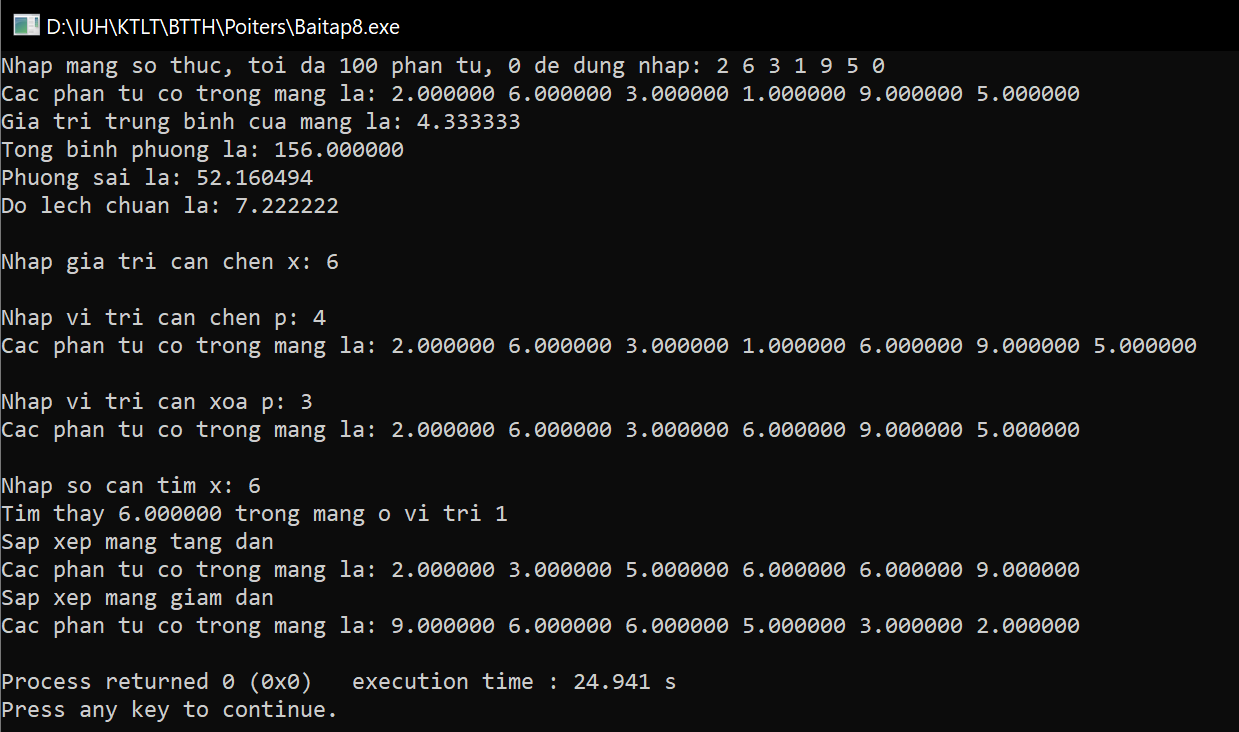
}

}

}

}

**Kết quả:**

****