**BỘ CÔNG THƯƠNG**

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



**LAB REPORT EXERCISES**

**KỸ THUẬT LẬP TRÌNH**

## MSSV: Họ Tên:

**NỘI DUNG 1: CON TRỎ**

**Câu 1:** Chương trình thực hiện các xử lý cơ bản trên con trỏ.

**Code:**

#include<stdio.h>

int main(){

int \*a;

int z;

z=10;

printf ("How works pointer and how to handle pointers:\n\n");

printf ("Value of z= %d\n",z);

printf ("Address of z=%p\n",&z);

a = &z;

printf ("\nNow a is assigned with the address of z:\n");

printf ("Address of pointer a= %p\n",a);

printf ("content of pointer a= %d\n\n",\*a);

printf ("Now, The value of z assigned to 30: \n");

printf ("Address of pointer a= %p\n",a);

printf("Content of pointer a= %d\n\n",\*a);

\*a = 5;

printf ("Now, pointer variable a is assigned the value 5\n");

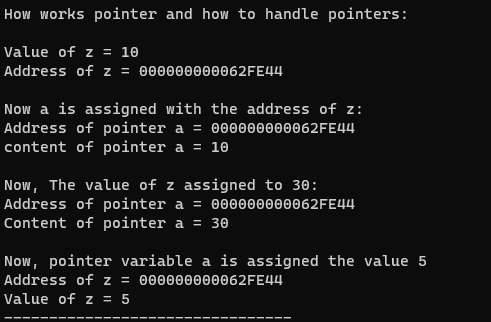
printf ("Address of z =%p\n",&z);

printf("Value of z=%d ", z);

return 0;

}

**Kết quả:**



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

**Code:**

#include <stdio.h>

int main(){

int a, b;

scanf ("%d%d", &a, &b);

int \*p1 = &a, \*p2 = &b;

printf ("%d\n", \*p1 + \*p2);

printf ("%d\n", \*p1 - \*p2);

printf ("%d\n", \*p1 \* \*p2);

printf ("%f\n", 1.0 \* \*p1 / \*p2);

}

**Kết quả:**

A screen shot of a computer

Description automatically generated

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

**Code:**

#include <stdio.h>

void max (int \*a, int \*b){

if (\*a > \*b)

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

else if (\*a < \*b)

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

else

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

}

int main(){

int a, b;

scanf ("%d%d", &a, &b);

max(&a, &b);

}

**Kết quả:**

A screenshot of a computer

Description automatically generated

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

**Code:**

#include <stdio.h>

void swap (int \*a, int \*b){

int tmp = \*a;

\*a = \*b;

\*b = tmp;

}

int main(){

int a, b;

scanf ("%d%d", &a, &b);

swap(&a, &b);

printf ("%d %d", a, b);

}

**Kết quả:**

A screenshot of a computer

Description automatically generated

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 máng

**Code:**

#include<stdio.h>

#include<stdlib.h>

//a

void input(int \*p, int n){

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

scanf("%d",p+i);

}

}

//b

void output(int \*p, int n){

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

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

}

printf("\n");

}

//c

void even(int \*p , int n){

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

if(p[i] % 2 == 0){

printf(" %d ",p[i]);

}

}

printf("\n");

}

void maxx(int \*p , int n){

int max = 0 ;

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

if(p[i] > max){

max = p[i];

}

}

printf("max = %d ",max);

}

int main(){

int \*p;

int n ;

p = (int\*)malloc(n\*sizeof(int));

scanf("%d",&n);

//a

input(p,n);

//b

output(p, n);

//c

even(p,n);

//g

maxx(p ,n);}

**Kết quả:**

A screen shot of a computer

Description automatically generated

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<math.h>

#include<stdio.h>

#include<stdlib.h>

int main(){

double \*p1, \*p2;

p1 = (double\*) malloc (sizeof(double));

p2 = (double\*) malloc (sizeof(double));

printf("p1, address: %u, value: %u\n", &p1, p1);

printf("p2, address: %u, value: %u\n", &p2, p2);

printf("Input 2 numbers:");

scanf("%lf%lf", p1, p2);

printf("Sum: %lf\n", \*p1 + \*p2);

printf("Difference: %lf\n", \*p1-\*p2);

printf("Product: %lf\n", (\*p1)\*(\*p2));

printf("Quotient: %lf\n", (\*p1)/(\*p2));

}

**Kết quả**

**A computer screen shot of a black screen

Description automatically generated**

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 \*p1, char \*p2){

int tmp = \*p1;

\*p1 = \*p2;

\*p2 = tmp;

}

int main(){

char \*p1, \*p2;

p1 = (char \*) malloc (sizeof(char));

p2 = (char \*) malloc (sizeof(char));

scanf ("%c%c", p1, p2);

if (\*p1 > \*p2)

swap(p1, p2);

for (char c = \*p1; c <= \*p2; c++){

printf ("%c %4d %4d %4X\n", c, c, c + 16, c);

}

free(p1);

free(p2);

}

**Kết quả**

**A screen shot of a computer

Description automatically generated**

Câu 8. Viết chương trình thực hiện các thao tác trên mảng 1 chiều, sử sụng kỹ thuật cấp

phát động (HEAP).

a) Nhập máng số thực.

b) Xuất các phần tử của màng.

c) Tính giá trị trung bình của mảng: m = (X + X1 X2+...+X-1)/n

d) Tính tổng bình phương: ss = x² + x1 + x2²+ ... + Χρ.2 e) Tính phương sai (Variance): d²= (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 vị trí P (P nhập từ bàn phím)

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

j) Sắp xếp màng tăng/giảm

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

//a

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

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

printf("Nhap phan tu thu %d: ", i);

scanf("%f", &a[i]);

}

//b

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

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

printf("%f ", a[i]);

}

printf("\n");

//c

float tinhGiaTriTrungBinh(float \*a, int n) {

float tong = 0;

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

tong += a[i];

}

return tong / n;

}

//d

float tinhTongBinhPhuong(float \*a, int n) {

float tong = 0;

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

tong += a[i] \* a[i];

}

return tong;

}

//e

float tinhPhuongSai(float \*a, int n) {

float m = tinhGiaTriTrungBinh(a, n);

float ss = tinhTongBinhPhuong(a, n);

return (ss / n) - m \* m;

}

//f

float tinhDoLechChuan(float \*a, int n) {

return sqrt(tinhPhuongSai(a, n));

}

//g

void themGiaTri(float \*a, int \*n, float x, int p) {

for(int i = \*n; i > p; i--) {

a[i] = a[i - 1];

}

a[p] = x;

(\*n)++;

}

//h

void xoaPhanTu(float \*a, int \*n, int p) {

for(int i = p; i < \*n - 1; i++) {

a[i] = a[i + 1];

}

(\*n)--;

}

//i

int timKiem(float \*a, int n, float x) {

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

if(a[i] == x) {

return i;

}

}

return -1;

}

//j

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

for(int i = 0; i < n - 1; i++) {

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

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

float temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

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

for(int i = 0; i < n - 1; i++) {

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

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

float temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

int main() {

int n;

printf("Nhap so luong phan tu: ");

scanf("%d", &n);

float \*a = (float\*)malloc(n \* sizeof(float));

//a

nhapMang(a, n);

//b

xuatMang(a, n);

//c

printf("Gia tri trung binh: %f\n", tinhGiaTriTrungBinh(a, n));

//d

printf("Tong binh phuong: %f\n", tinhTongBinhPhuong(a, n));

//e

printf("Phuong sai: %f\n", tinhPhuongSai(a, n));

//f

printf("Do lech chuan: %f\n", tinhDoLechChuan(a, n));

//g

float x;

int p;

printf("Nhap gia tri can them: ");

scanf("%f", &x);

printf("Nhap vi tri can them: ");

scanf("%d", &p);

themGiaTri(a, &n, x, p);

xuatMang(a, n);

//h

printf("Nhap vi tri can xoa: ");

scanf("%d", &p);

xoaPhanTu(a, &n, p);

xuatMang(a, n);

//i

printf("Nhap gia tri can tim: ");

scanf("%f", &x);

int viTri = timKiem(a, n, x);

if(viTri == -1) {

printf("Khong tim thay gia tri trong mang.\n");

} else {

printf("Gia tri tim thay tai vi tri: %d\n", viTri);

}

//j

sapXepTang(a, n);

printf("Mang sau khi sap xep tang: ");

xuatMang(a, n);

sapXepGiam(a, n);

printf("Mang sau khi sap xep giam: ");

xuatMang(a, n);

free(a);

return 0;

}

**Kết quả**

**A screenshot of a computer

Description automatically generated**