**Bài kiểm tra cuối môn**

**Lập trình cơ bản**

Thời gian làm bài: 2.15 h

**Bài 1 (1đ)**

Viết chương trình để làm các việc sau:

- Khai báo biến a kiểu số nguyên, được gán giá trị 10

- Khai báo biến b kiểu số thực có độ chính xác kép, được gán giá trị 100.5E+2

- Khai báo biến c kiểu ký tự được gán giá trị 'x'

- In các biến a, b, c ra ngoài màn hình, mỗi biến trên một dòng

**Bài 2 (1.2 đ)**

Viết chương trình để làm các việc sau:

- Khai báo mảng số nguyên 10 phần tử, được gán sẵn các giá trị trong đó.

- In các phần tử trong mảng

- In các phần tử chẵn trong mảng

- In các phần tử chia hết cho 3 hoặc cho 5

**Bài 3 (0,8đ)**

Khai báo nguyên mẫu hàm:

1. Hàm in một mảng số thực có N phần tử

2. Hàm tính căn bậc hai của tổng N phần tử

3. Hàm hoán đổi giá trị của hai biến ký tự x và y cho nhau

4. Tính tổng của mảng số nguyên A có N phần tử và mảng số thực B có M phần tử

5. Tính căn bậc x của n

6. Kiểm tra một số nguyên là dương hay âm

7. Tìm số bé nhất trong một mảng số thực

8. Tìm vị trí số x trong mảng số nguyên n phần tử (Lấy vị trí đầu tiên xuất hiện nếu các phần tử trùng nhau). Nếu không tìm thấy thì hiển thị "Not Found!"

**Bài 4 (0.5 đ)**

Để khai báo mảng số thực có độ chính xác kép A có 5 phần tử, phương án đúng là:

A. int A[5];

B. char A[5];

C. double A[5];

D. int A[] = {5};

**Bài 5 (0.5 đ)**

Cho biết kết quả đoạn chương trình sau:

void main() {

int a = 4;

if (a % 1 == 0)

printf("1");

else if (a % 2 == 0)

printf("2");

if (a % 3 == 0)

printf("3");

else if (a % 4 == 0)

printf("4");

if (a % 5 == 0)

printf("5");

else

printf("6");

\_getch();

}

A. 1

B. 12

C. 14

D. 146

**Bài 6 (0.5 đ)**

Cho biết kết quả của chương trình sau:

#include <stdio.h>

int main()

{

int a = 10;

if (a == a--)

printf("TRUE 1 \t");

a = 10;

if (a == --a)

printf("TRUE 2 \t");

}

A - TRUE 1

B - TRUE 2

C - TRUE 1 TRUE 2

D - Không có kết quả

**Bài 7 (0.5 đ)**

Cho biết kết quả của chương trình sau:

void main() {

int n = 8;

int tong;

int a[] = { 1, 2, 5, 7, -4, 4, 9, -8 };

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

{

tong += a[i];

}

printf("Tong la: %d ", tong);

\_getch();

}

A. 16

B. Lỗi cú pháp

C. Lỗi biên dịch

D. 24

**Bài 8 (2đ)**

Viết chương trình nhập vào mảng số nguyên dương gồm N phần tử. Hãy tính tổng các số đối xứng trong mảng. Biết rằng số đối xứng là số viết xuôi hay ngược vẫn là chính nó.

Ví dụ: 11,121…

Chương trình bắt buộc chia thành tối thiểu 3 hàm như sau:

- Hàm nhập liệu

- Hàm hiển thị kết quả

- Hàm xử lý (Lưu ý chia nhỏ các công việc trong cần xử lý thành các hàm con nếu có thể)

**Bài 9 (1đ)**

Cho mảng ký tự arr, trong mảng này có lẫn cả ký tự số và ký tự không phải số.

Hãy tính tổng các số được tạo thành trong mảng, in kết quả này ra dưới dạng các phép tính “+” và ghi kết quả sau dấu bằng “=”

VD:

arr = {‘a’,’b’,’1’,’c’,’2’,’3’,’d’,’4’,’5’,’e’}

In ra 1 + 23 +45 = 69

Bài 10 **(2 điểm)**

Viết chương trình nhập vào mảng số nguyên dương gồm N phần tử. Tính tổng nghịch đảo các giai thừa của tổng ước số mỗi phần tử không tính chính nó trong mảng.

Ví dụ: A[] = {4, 12, 6, 8, 10};

=> Cần tính S = 1/3! + 1/16! + 1/6! + 1/7! + 1/8!

Với 3, 16, 6, 7, 8 lần lượt là tổng các ước số của 4, 12, 6, 8, 10

Chương trình bắt buộc chia thành tối thiểu 3 hàm như sau:

- Hàm nhập liệu

- Hàm hiển thị kết quả

- Hàm xử lý (Lưu ý chia nhỏ các công việc trong cần xử lý thành các hàm con nếu có thể)

**Chúc các bạn làm bài tập tốt!**

**Bài làm**

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <ctype.h>

#include <stdio.h>

#include <stdbool.h>

#include <string.h>

#include <iostream>

//Cau1

//-Khai báo biến a kiểu số nguyên, được gán giá trị 10

//- Khai báo biến b kiểu số thực có độ chính xác kép, được gán giá trị 100.5E+2

//- Khai báo biến c kiểu ký tự được gán giá trị 'x'

//- In các biến a, b, c ra ngoài màn hình, mỗi biến trên một dòng

void Cau1() {

int a = 10;

double b = 100.5E+2;

char c = 'x';

printf("a = %d\nb = %lf\nc = %c", a, b, c);

}

//Viết chương trình để làm các việc sau :

//-Khai báo mảng số nguyên 10 phần tử, được gán sẵn các giá trị trong đó.

//- In các phần tử trong mảng

//- In các phần tử chẵn trong mảng

//- In các phần tử chia hết cho 3 hoặc cho 5

void Cau2() {

int a[] = { 1,2,3,4,5,6,7,8,9,10 };

int n = 10;

printf("\nMang da cho la: ");

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

{

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

}

printf("\nCac phan tu chan la: ");

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

{

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

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

}

}

printf("\nCac phan tu chia het cho 3 hoac 5 la: ");

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

{

if (a[i] % 3 == 0 || (a[i] % 5 == 0)) {

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

}

}

}

// Cau 3

//Khai báo nguyên mẫu hàm :

////1. Hàm in một mảng số thực có N phần tử

void inMangThuc(double arr[], int n);

////2. Hàm tính căn bậc hai của tổng N phần tử

double sqrtOfSum(int arr[], int n);

////3. Hàm hoán đổi giá trị của hai biến ký tự x và y cho nhau

void hoanVi(int& x, int& y);

////4. Tính tổng của mảng số nguyên A có N phần tử và mảng số thực B có M phần tử

int sumA();

void sumB(int arr[], int n, double arrb[], int m);

////5. Tính căn bậc x của n

double sqrtOfNum(int n);

////6. Kiểm tra một số nguyên là dương hay âm

bool checkNum(int n);

////7. Tìm số bé nhất trong một mảng số thực

int minArr(double arr[], int n);

////8. Tìm vị trí số x trong mảng số nguyên n phần tử(Lấy vị trí đầu tiên xuất hiện nếu các phần tử trùng nhau).Nếu không tìm thấy thì hiển thị Not Found!

int timX(int arr[], int n, int x);

//Cau 4 C

//Cau 5 D

//Cau 6 A

//Cau 7 B

//Cau 8

void nhapMang(int a[], int n){

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

printf("\na[%d] = ", i);

scanf\_s("%d", &a[i]);

}

}

bool getSoDoiXung(int n) {

int numdao = 0, rem;//numdao so dao nguoc, rem so du

int tmp = n;

while (tmp != 0)

{

// Lấy chữ số cuối của số nguyên sử dụng toán tử chia lấy dư (Modulus)

rem = tmp % 10;

// Nhân số đảo với 10 và cộng với chữ số cuối

numdao = (numdao \* 10) + rem;

// Xóa chữ số cuối bằng cách sử dụng toán tử chia lấy phần nguyên (Division)

tmp = tmp / 10;

}

if (numdao == n && numdao > 9) {

return true;

}

return false;

}

int sumNumDoiXung(int arr[], int n) {

int sum = 0;

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

{

if (getSoDoiXung(arr[i]) == true) {

sum += arr[i];

}

}

return sum;

}

//Cau 9

//Cho mảng ký tự arr, trong mảng này có lẫn cả ký tự số và ký tự không phải số.

//Hãy tính tổng các số được tạo thành trong mảng, in kết quả này ra dưới dạng các phép tính “ + ” và ghi kết quả sau dấu bằng “ = ”

//VD :

//arr = { ‘a’,’b’,’1’,’c’,’2’,’3’,’d’,’4’,’5’,’e’ }

//In ra 1 + 23 + 45 = 69

void nhapChuoi(char s[]) {

printf("Nhap chuoi s: ");

fgets(s, 100, stdin);

}

int strLength(char s[]) {

int i = 0, length = 0;

while (s[i] != '\0') {

length = i;

i++;

}

return length;

}

bool isNumber(char c);

bool isNumber(char c) {

return (c >= '0' && c <= '9'); // xet bang ma ascii

}

int ghepNum(int socu, char kt) {

return socu = socu \* 10 + (kt - '0');

}

int sumMatchNum(char \*s) {

int n = strLength(s), sum = 0, num = 0;

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

{

if (isNumber(s[i])) {

num = ghepNum(num, s[i]);

}

else

{

sum += num;

num = 0;

}

}

if (isNumber(s[n - 1])) {

sum += num;

}

return sum;

}

// Cau 10

//Viết chương trình nhập vào mảng số nguyên dương gồm N phần tử.Tính tổng nghịch đảo các giai thừa của tổng ước số mỗi phần tử không tính chính nó trong mảng.

//Ví dụ : A[] = { 4, 12, 6, 8, 10 };

//= > Cần tính S = 1 / 3!+ 1 / 16!+ 1 / 6!+ 1 / 7!+ 1 / 8!

//Với 3, 16, 6, 7, 8 lần lượt là tổng các ước số của 4, 12, 6, 8, 10

//Chương trình bắt buộc chia thành tối thiểu 3 hàm như sau :

//-Hàm nhập liệu

//- Hàm hiển thị kết quả

//- Hàm xử lý(Lưu ý chia nhỏ các công việc trong cần xử lý thành các hàm con nếu có thể)

int getSumUOC(int n) {

int sum = 0;

for (int i = 1;i <= n / 2;i++) {

if (n % i == 0) {

sum += i;

}

}

return sum;

}

int giaiThua(int n)

{

int res = 1;

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

res \*= i;

}

return res;

}

double sumGiaiThuaDao(int a[], int n) {

double sum = 0;

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

{

sum += (1.0 / (giaiThua(getSumUOC(a[i]))));

}

return sum;

}

void main() {

int n;

int a[100];

char s[100];

Cau1();

Cau2();

//Cau 8

/\*printf("\nNhap n phan tu: ");

scanf\_s("%d", &n);

while (n < 0 || n>100) {

printf("\nKhong hop le, nhap lai: ");

scanf\_s("%d", &n);

}

nhapMang(a, n);

printf("\nTong cac so doi xung la: %d", sumNumDoiXung(a, n));\*/

//Cau 9

nhapChuoi(s);

int sum = sumMatchNum(s);

printf("\nTong cac so trong day ky tu = %d ", sum);

//Cau 10

//printf("\nNhap n phan tu: ");

// scanf\_s("%d", &n);

// while (n < 0 || n>100) {

// printf("\nKhong hop le, nhap lai: ");

// scanf\_s("%d", &n);

// }

// nhapMang(a, n);

// double sum = sumGiaiThuaDao(a, n);

// printf("\nTong cac nghich dao cua giai thua la tong cac uoc cua moi phan tu: %lf", sum);

\_getch;

}