

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA KHOA HỌC MÁY TÍNH	Đề thi cuối kỳ: Nhập môn lập trình Học kỳ 1 - Năm học 2016-2017 Thời gian: 90 phút	Điểm số
Chữ ký cán bộ coi thi 1: ..... Chữ ký cán bộ coi thi 2: .....	STT: ..... MSSV: ..... Họ Tên: .....	

**Lưu ý:** Sinh viên làm bài trực tiếp trên đề, không được sử dụng tài liệu.

**Câu 1:** Cho đoạn chương trình sau (0.5 điểm):

```
#include <iostream>
using namespace std;
void hamf(int x, int &y)
{ x = x+y;
  y = x-y;
  x = x-y;
}
int main()
{   int a=1,b=2;
    hamf(a,b);
    cout<<"a = "<<a<<endl;
    cout<<"b = "<<b;
    return 0;
}
```

Sau khi thực hiện xong chương trình. Kết quả in ra : a = 1 ..... b = 1....

→ Mỗi đáp án đúng là 0.25 điểm

**Câu 2:** Cho các dòng khai báo sau (1.75 điểm)::

Dòng lệnh số	Chương trình
1.	int size; double list[size];
2.	double a[40.5];
3.	char sName[] = "Johnson", s[100] ;
4.	s = sName;
5.	char s1[] = "An", s2[] = "Apple", s3[100];
6.	s3 = s1+s2;
7.	int *ptr, x = 1;
8.	ptr = &(x+5);
9.	int d[5]={1,2,3,4,5};
10.	d++;
11.	int *p;
12.	*p=2;

Hãy cho biết dòng lệnh nào **sai** (về cú pháp hoặc logic), và cho biết **tại sao sai** ?

Mỗi đáp án đúng là 0.25 điểm. Có 7 đáp án → 1.75 điểm. Sau đây là đáp án gợi ý:

- Dòng 1 sai, do kích thước của mảng cố định khi khai báo không được là giá trị của một biến.
- Dòng 2 sai, do kích thước của mảng cố định khi khai báo không được là giá trị số thực.

- Dòng 4 sai, do không thể thực hiện phép gán cho biến s, phải dùng hàm strcpy
- Dòng 6 sai, do không thể thực hiện toán tử + giữa hai biến s1 và s2.
- Dòng 8 sai, do không thể lấy địa chỉ của một biểu thức.
- Dòng 10 sai, do biến d là một biến con trỏ hằng nên không được thay đổi.
- Dòng 12 sai, do con trỏ p có giá trị không xác định, nên thao tác \*p bị lỗi.

**Câu 3:** Hãy viết các lệnh còn thiếu để hoàn thành chương trình sau (0.75 điểm):

```
int main()
{
    char str1[200];
    char str2[200];
    // Lệnh để gán chuỗi "Truong Dai hoc" vào biến str1 :
    strcpy(str1, "Truong Dai hoc"); .....
    // Lệnh để gán chuỗi "Cong nghe Thong tin" vào biến str2 :
    strcpy(str2, "Cong nghe thong tin"); .....
    → Hai lệnh trên: 0.25 điểm.
    // Đo chiều dài của chuỗi str1 và lưu trữ vào biến length:
    int length = strlen(str1); → 0.25 điểm
    cout<<"Chuoi:"<< str1<<" co "<<length<<" ky tu"<< endl;
    // So sánh và in ra màn hình chuỗi nhỏ nhất:
    cout<<"Chuoi nho nhat la:"
    if(strcmp(str1, str2)<0) → 0.25 điểm
        cout<<str1;
    else cout<<str2;
    return 0;
}
```

**Câu 4:** Hãy viết các lệnh trong hàm TinhTong để tính tổng hai phân số và cho biết kết quả của chương trình sau (0.75 điểm):

```
#include <iostream>
using namespace std;
struct PhanSo
{
    int TuSo;
    int Mauso;
}; // cấu trúc PhanSo để mô tả các phân số

PhanSo TinhTong (PhanSo x, PhanSo y)
{
    PhanSo tong;
    tong.TuSo = x.TuSo*y.Mauso+y.TuSo*x.Mauso;
    tong.Mauso = x.Mauso*y.Mauso;
    return tong;
}
```

→ viết đúng hàm : 0.5 điểm

```

int main()
{
    PhanSo x,y,S;
    x.TuSo = 1; x.Mauso = 2;
    y.TuSo = 3; y.Mauso = 4;

    S=TinhTong(x,y);
    cout<<"Phan so S:";
    cout<<S.TuSo<<"/"<<S.Mauso;

    return 0;
}

```

Kết quả của đoạn chương trình trên là:

Phan so S: **10/8** → 0.25 điểm

**Câu 5:** Cho đoạn chương trình sau (1.25 điểm):

```

#include <iostream>
using namespace std;
int main()
{
    int a=3200;
    int *ptr;
    int value;
    ptr = &a;
    value = --(*ptr);
    return 0;
}

```

Giả sử khi chương trình thực thi, biến a được cấp phát bộ nhớ có địa chỉ 0x20, biến ptr được cấp phát bộ nhớ có địa chỉ 0x32, biến value được cấp phát bộ nhớ có địa chỉ 0x45. Hãy cho biết khi chương trình thực hiện đến lệnh return 0.

Giá trị của biến value là: <b>3199</b> .	Giá trị của &value là: <b>0x20</b> ....	Giá trị của *ptr là: <b>3199</b> ....
Giá trị của biến ptr là: <b>0x20</b> ...	Giá trị của &ptr là: <b>0x32</b> .....	Giá trị của &(*ptr) là: <b>0x20</b> .
Giá trị của biến a là: <b>3199</b> .....	Giá trị của *(*ptr) là: <b>sai/error</b> .	

→ Mỗi đáp án đúng là 0.125 điểm

→ Đáp án Giá trị của biến ptr là 0x20 và Giá trị của biến a là: 3199, mỗi đáp án là 0.25

**Câu 6:** Cho đoạn chương trình sau (0.5 điểm):

```

#include <iostream>
using namespace std;
int main()
{
    int *a = new int[5];
    for(int i=0; i<5; i++) a[i] = i+1;
    int *p=a ;
    cout<<"Gia tri *(p+2):"<< *(p+2);
    p+=2;
    cout<<"Gia tri *p:"<< *p;
    delete []a;
    return 0;
}

```

Kết quả của đoạn chương trình trên là:

- Gia trị \*(p+2): **3**.....
- Gia trị \*p: **3**.....

→ Mỗi đáp án đúng là 0.25 điểm

**Câu 7:** Cho đoạn chương trình sau (0.5 điểm):

```
#include <iostream>
using namespace std;
int kiemtradoixung(int a[], int n)
{
    for(int i=0; i<n/2; i++)
    {
        if( a[i]!=a[n-i-1]) return 0;
        else return 1;
    }
}
int main()
{
    int a[5] = {3, 2, 1, 5, 3};
    int n=5;
    if(kiemtradoixung(a,n)==1)
        cout<<"Mang doi xung";
    else cout<<"Mang khong doi xung";
}
```

Tìm lỗi sai của hàm kiemtradoixung và chỉnh sửa lại cho đúng.

**Đáp án gợi ý:** Câu này sinh viên có thể viết lại hàm kiemtradoixung theo nhiều cách.

**Cách đơn giản như sau:** .....

```
int kiemtradoixung(int a[], int n)
{
    for(int i=0; i<n/2; i++)
    {
        if( a[i]!=a[n-i-1]) return 0;
        else return 1;
    }
    return 1;
}
```

→ Sinh viên làm đúng được 0.5 điểm

**Câu 8:** (0.5 điểm)

Cho một mảng hai chiều beta[3][3] và đoạn mã như sau:

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = 2 * (i + j) % 4;
```

Hãy cho biết giá trị của các phần tử của mảng beta sau khi thực hiện đoạn mã trên? (điền kết quả vào hình bên dưới)

0	2	0
2	0	2
0	2	0

→ Sinh viên điền đúng được 0.5 điểm

**Câu 9: Cho hàm main sau (được 2.5 điểm):**

```
int main()
{   int a[50], n=0;
    int S, k,maxvalue;
    Nhapmang(a,n);
    S= TinhTongSNT(a,n);
    do{
        cout<<"Nhap gia tri k=";
        cin>>k;
    }while(k<1 || k >n);
    maxvalue = GiaTriLonThuK(a,n,k);
    cout<<"Gia tri lon thu"<<k<<" la:" << maxvalue;
    return 0;
}
```

Yêu cầu sinh viên viết các hàm sau:

- a) Hàm nhập mảng số nguyên a có n phần tử từ bàn phím (Nhapmang(a,n)) (0.75 điểm)

Nội dung hàm	Điểm
<code>void Nhapmang(int a[], int &amp;n)</code>	0.25
<pre>{   //cout&lt;&lt;"Nhap so phan tu n=";     cin&gt;&gt;n;     for(int i=0; i&lt;n; i++)     {       //cout&lt;&lt;"Nhap gia tri a["&lt;&lt;i&lt;&lt;"]=";             cin&gt;&gt;a[i];     } }</pre>	0.5

→ Sinh viên viết đúng được 0.75 điểm

→ Nếu không truyền tham biến (int &n) thì trừ 0.25

Lưu ý: do biến n được khởi tạo trong hàm main là 0, nên hàm này sinh viên phải nhập n và tham số n phải là tham chiếu.

- b) Hàm tính tổng các phần tử có giá trị là số nguyên tố dương (TinhTongSNT(a,n)) (1.25 điểm)

Câu này sinh viên có thể làm theo nhiều cách, có thể kiểm tra số nguyên tố trực tiếp hoặc viết thành hàm riêng. Đáp án gợi ý như sau:

Nội dung hàm kiểm tra 1 số có phải là số nguyên tố dương ?	Điểm
<pre>int kiemtrasonguyento(int n) {   if(n&lt;2) return 0;</pre>	0.25

<pre> for(int i=2; i&lt;n; i++) {     if(n%i==0)         return 0; } return 1; } </pre>	0.5
---	-----

→ Sinh viên viết đúng hàm hoặc thao tác kiểm tra số nguyên tố dương sẽ được 0.75 điểm

→ Nếu thiếu kiểm tra điều kiện nguyên tố dương ( $n < 2$ ) thì trừ 0.25 điểm

Nội dung hàm	Điểm
<pre> int TinhTongSNT (int a[], int n) {     int S=0;     for(int i=0; i&lt;n; i++)     {         if(kiemtrasonguyento(a[i])==1)             S = S+a[i];     }     return S; } </pre>	0.5

→ Sinh viên viết đúng hàm sẽ được 0.5 điểm

- c) Giả sử các phần tử của mảng khác nhau đôi một. Hãy viết hàm tìm giá trị lớn thứ k ( $1 \leq k \leq n$ ) trong mảng ( $\text{GiaTriLonThuK}(a, n, k)$ ) (0.5 điểm)

Ví dụ: Cho mảng  $a = \{3, 7, 9, 6, 1, 5, 2\}$ , giá trị lớn thứ 2 trong mảng là 7, giá trị lớn thứ 4 trong mảng là 5.

Câu này là một câu hỏi mở rộng, sinh viên có thể có nhiều cách làm khác nhau. Sau đây là đáp án gợi ý:

Cách 1: Cách đơn giản

Sắp xếp mảng đã cho giảm dần rồi return về phần tử  $a[k-1]$  của mảng (đã được sắp thứ tự)

Hoặc là sắp xếp mảng tăng dần và return về phần tử  $a[n-k]$

```

int GiaTriLonThuK(int arr[], int n, int k)
{
    // Sắp mảng theo thứ tự giảm dần
    sort(arr, n);
    // Trả về phần tử thứ k
    return arr[k-1];
}

```

**Cách 2: Ý tưởng như Bubble Sort, thực hiện vòng lặp ngoài k lần để đưa phần tử lớn thứ k về vị trí n-k trong mảng**

```
int GiaTriLonThuK(int arr[], int n, int k)
{
    int temp;
    // Modify Bubble Sort to run the outer loop k times.
    for (int i = 1; i <= k; i++)
        for (int j = 0; j < n-i; j++)
            if (arr[j] > arr[j+1])
                { temp = arr[j]; arr[j]=arr[j+1];arr[j+1] = temp;}

    // Return k'th largest element in the array
    return arr[n-k];
}
```

**Cách 3: Ý tưởng như Selection Sort, tìm phần tử lớn nhất k lần**

```
int GiaTriLonThuK(int arr[], int n, int k)
{
    int temp, maxpos;
    // Modify Selection Sort to run the outer loop k times.
    for (int i = 0; i < k; i++)
    {
        maxpos = i;
        for (int j = i+1; j < n; j++)
            if (arr[j] > arr[maxpos])
                maxpos = j;
        temp = arr[i]; arr[i]=arr[maxpos];arr[maxpos] = temp;
    }

    return arr[k-1];
}
```

**Cách 4: Tìm số lớn nhất trong mảng, xóa số đó ra khỏi mảng và lặp lại như vậy k lần**

**Câu 10:** Sinh viên thực hiện các yêu cầu sau (1.0 điểm)

a) Khai báo kiểu dữ liệu cấu trúc (struct) Sinhvien gồm các thuộc tính (thành phần) sau (0.5 điểm)

Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
MaSV	Kiểu chuỗi ký tự, chiều dài tối đa 8 ký tự.	Mã sinh viên
HoTen	Kiểu chuỗi ký tự, chiều dài tối đa 250 ký tự.	Họ và tên sinh viên
Namsinh	Kiểu số nguyên	Năm sinh
DTB	Số thực	Điểm trung bình

struct Sinhvien

```
{ char MaSV[9]; // sinh viên khai báo char MaSV[8]; cũng chấp nhận được.  
  char HoTen[251]; // sinh viên khai báo char Hoten[250]; cũng chấp nhận được.  
  int Namsinh;  
  float DTB; // double DTB;  
};
```

→ Sinh viên khai báo đúng được 0.5 điểm

b) Viết hàm để tìm kiếm một sinh viên trong mảng sinh viên dựa trên thuộc tính mã số sinh viên. Nếu tìm thấy, hàm trả về vị trí của sinh viên trong mảng. Nếu không tìm thấy, hàm trả về kết quả -1. (0.5 điểm)

Nội dung hàm tính tồn	Điểm
<pre>int TimSV(Sinhvien a[], int n, char x[]) // hoặc char *x</pre>	0.25
<pre>{ for(int i=0; i&lt;n; i++)     if( strcmp(a[i].MaSV,x)==0)         return i;     return -1; }</pre>	0.25

----- Hết -----