

ĐỀ 2

MÃ ĐỀ: Tôi ngay thẳng

Môn thi: Nhập môn lập trình

Mã môn/lớp: IT001.

Thời gian làm bài: 90 phút

(Sinh viên không được sử dụng tài liệu)

Thang điểm			
Câu 1: Phép toán số học	0.5	Câu 9: Mảng 1 chiều: kiểm tra tính chất mảng	0.5
Câu 2: Phép toán khác	1	Câu 10: Chuỗi	0.75
Câu 3: Rẽ nhánh switch...case	0.5	Câu 11: Mảng 2 chiều: duyệt 1 phần của mảng	0.5
Câu 4: Vòng lặp do.while, continue	0.5	Câu 12: Mảng 2 chiều: kiểm tra tính chất mảng	0.5
Câu 5: Vòng lặp for	0.5	Câu 13: Con trỏ: nhận diện và giải thích được các lỗi (chú trọng kỹ năng diễn đạt)	1
Câu 6: Hàm: truyền tham số cho hàm	0.5	Câu 14: Con trỏ và mảng 1 chiều: hiểu ý nghĩa của từng câu lệnh (chú trọng kỹ năng diễn đạt)	0.75
Câu 7: Hàm: biến tham chiếu	0.5	Câu 15: Cấu trúc (kết hợp con trỏ, tổ chức các hàm)	0.75
Câu 8: Hàm đệ quy	0.5	Câu 16: Phân tích bài toán, sinh testcase	0.75

Câu 1: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main()
{
    cout<<13.0/5-24%2<<",";
    cout<<(int)-7.1*(21/6)+(float)7/4;
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án: có 2 số	Thang điểm	Ghi chú
2.6	-19.25	Đúng 1 số đạt 0.25đ -Không có dấu , ngăn cách cũng không trừ điểm, chỉ xét giá trị -Nhiều SV ghi nhầm (hoặc khó phân biệt) giữa dấu , và .

Câu 2: (1 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main()
{    int x=12, y=24;
    cout<<(y!=24 || (x<5&&y>=24))<<", ";
    cout<<(x>y?y-x:y/3)<<", ";
    cout<<(y>x>10)<<", ";
    cout<< (!0 && !(x==y));
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án: có 4 số				Thang điểm	Ghi chú
0	8	0	1	Đúng 1 số đạt 0.25đ	<ul style="list-style-type: none">- Nhiều SV ghi đáp án là “0, 8, ,0”, hay “0,8, ,” vẫn chấm theo vị trí của các số- Nhiều SV chỉ ghi 2-3 số mà không đủ 3 dấu , phân cách (VD “0,8”) thì 0đ

Câu 3: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main()
{    int x,y;
    cin >> x;
    y=x;
    switch (x)
    {    case 1:
        case 2:
            y *= y+x;
        case 4:
            y--;
        case 5:
            y = 2*x-10; break;
        case 6:
            y = y/5;   break;
        default:
            y+=2;
    }
    cout<<y;
    return 0;
}
```

Nếu nhập giá trị của biến x là **1** thì kết quả in ra màn hình là:

Nếu nhập giá trị của biến x là **3** thì kết quả in ra màn hình là:

Đáp án: có 2 số	Thang điểm
-8 5	Mỗi số đúng đạt 0.25đ

Câu 4: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main()
{
    int sum, num;
    cin>>sum;
    do
    {
        cin>>num;
        if(num%5!=0) continue;
        sum+=num;
    }
    while(sum < 100);
    cout<<"sum="<<sum;
    return 0;
}
```

Giả sử user muốn nhập lần lượt các số sau **11 5 2 30 9 10 15 -10 50 7 10** (có thể không thể nhập hết được dãy số này)

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án: có 2 trường hợp đạt điểm	Thang điểm
sum=111	0.5đ
111(Chỉ ghi duy nhất số 111)	0.25đ

Câu 5: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main ()
{
    int n=518, s=0;
    for(int i=n; i>0; i=i/10)
        s += i;
    cout<<s;
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án:	Thang điểm	Ghi chú
574	0.5đ	574 là 1 số. Bất kỳ 1 số nào khác đều là sai, 0đ VD, nhiều SV ghi “57” hay “575” là sai, 0đ

Câu 6: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
void fun(int &a, int b, int &c)
{
    a-= b*c;
    b = 5*c;
    c = 10;
}
int main()
{
    int x=1, y=5, z=4;
    fun(z, x, y);
    cout<<x<<y<<z;
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án: có 3 số			Thang điểm	Ghi chú
1	10	-1	-Chỉ sai 1 số: 0.25đ -Sai 2 số: 0đ	Chỉ có 3 mức điểm: 0, 0.25, 0.5, không chia nhỏ hơn nữa SV ghi 1,10,-1 cũng không sao, không trừ điểm

Câu 7: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;

int test(int a)
{
    a=a*2-7;
    return -a;
}
int main()
{
    int a=7;
    int &b=a;
    cout<<test(a)<<", ";
    b=100;
    a=b-a+1;
    b++;
    cout<<a<<", "<<b;
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là

Đáp án: có 3 số			Thang điểm	Ghi chú
-7	2	2	-Chỉ sai 1 số: 0.25đ -Sai 2 số: 0đ	-Chỉ có 3 mức điểm: 0, 0.25, 0.5, không chia nhỏ hơn nữa -Nhiều SV phân tách các số bằng khoảng trắng, không dùng dấu , thì cũng không trừ điểm

Câu 8: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int fun(int x)
{
    if(x<2) return 1;
    return (x+1)*fun(x-1);
}
int main () {
    cout<<fun(3)+fun(4);
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án:	Thang điểm	Ghi chú
72	0.5đ	Bất kỳ 1 số nào khác đều là sai, 0đ

Câu 9: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
bool KiemTra (int *a, int n){
    for (int i=1; i<n; i++)
        if (a[i] > a[i-1]) return false;
    return true;
}
int main()
{
    int n, a[100];
    cin>>n;
    for(int i=0;i<n;i++)
        cin>>a[i];
    cout<<KiemTra(a,n);
}
```

Với các trường hợp nhập dữ liệu đầu vào như ở cột Input dưới đây, hãy điền kết quả in ra màn hình ở cột Output.

Input	Output: có 2 cách ghi		Thang điểm	Ghi chú
1 7	1	true	0.25đ	Chấm theo cặp. Ở mỗi cặp, phải đúng hết cả 2 mới đạt 0.25 điểm, sai 1 số là 0đ
2 5 5	1	true		
5 3 3 4 4 5	0	false	0.25đ	VD, ở cặp thứ 2, nếu SV ghi (1 1) là 0đ, không chia điểm nhỏ hơn nữa
10 6 6 5 4 4 4 3 2 1 1	1	true		

Câu 10: (0.75 điểm) Hãy hoàn thành chương trình bên dưới để thực hiện việc đếm số lượng ký tự in thường trong một chuỗi.

```
#include <iostream>
using namespace std;
int count_LowerCase(char *s)

{    int i=0, dem=0;

    while(s[i]!='\0')          (1)
    {
        if('a'<=s[i]&&s[i]<='z') (2)
            dem++;

        i++;                      (3)
    }

    return dem;
}

int main()
{
    char str[35];
    cin.getline(str, 30);
    cout<<count_LowerCase(str);
}
```

Vị trí	Đáp án	Đáp án khác	Thang điểm	Ghi chú: Những trường hợp sau là sai, 0đ
(1)	s[i]!='\0'	*(s+i)!= '\0' s[i]!=NULL	0.25đ	SAI: KHÔNG CÓ ĐIỂM i<strlen, i<strlen(s) i<=strlen(s) (không có thư viện) i>strlen(s) i<30, i<=30 (sẽ truy cập giá trị rác) s[i]≠ '\0' i! ="\0'" s[i]!= 'NULL', s[i]!= 'null'
(2)	'a'<=s[i]&&s[i]<='z'	'a'<=s[i]ands[i]<='z' 97<=s[i]&&s[i]<=122 'a'<=*(s+i)&&*(s+i)<= 'z'	0.25đ	SAI: KHÔNG CÓ ĐIỂM 'a'<=s[i] (chỉ so với a hoặc chỉ so với z)

	(s[i]=='a' s[i]=='z' ('a'<s[i]&&s[i]<'z'))		'A'<=s[i]&&s[i]<='Z' (chữ hoa) 'a'<=s[i]<='z' (không có &&) "a"<=s[i]&&s[i]<="z" (dùng nháy kép, không phải ký tự) a<=s[i]&&s[i]<=z (thiếu dấu nháy đơn, không phải ký tự) 'a'<=s[i] s[i]<='z' (sai dấu) 'a'<s[i]&&s[i]<'z' (thiếu =) s[i]='a' to 'z' s[i]==tolower(s[i]) (đè không gọi thư viện cstring) s[i]=strlwr(s[i]) (đè không gọi thư viện)	
(3)	i++; return dem;		0.25đ	Đúng hết cả 2 mới đạt 0.25đ, sai/thiếu 1 trong 2 là 0đ
	Nhiều SV dùng sai tên chuỗi, thay vì s[i] thì lại gọi a[i], str[i] thì TRỪ 0.25đ			

Câu 11: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int main()
{    int a[5][5]={ {1,0,3,2,1},
                 {1,8,2,0,5},
                 {7,2,3,1,4},
                 {4,3,0,1,3},
                 {1,2,3,4,5}};

    int S=0;
    for(int i=2; i<5;i++)
        for(int j=1;j<4;j++)
            if(i!=j)
                S+=a[i][j];
    cout<<S;
    return 0;
```

}

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án:	Thang điểm	Ghi chú
15	0.5đ	Bất kỳ 1 số nào khác đều là sai, 0đ

Câu 12: (0.5 điểm) Cho một chương trình như sau:

```
#include <iostream>
using namespace std;
int fun(int arr[][][6], int n)
{
    int i,j;
    for (i = 0; i < n; i++)
    {
        for (j=0;j<n;j++)
            if(arr[i][j] > 0)
                break;
        if (j==n) return i;

        for (j=0;j<n;j++)
            if(arr[i][j] < 0)
                break;
        if (j==n) return i;
    }
    return -1;
}
int main()
{
    int a[6][6]={ {0,-1,2,-3,4,-5},
                  {-1,0,-1,0,-1,0},
                  {0,1,0,1,0,1},
                  {1,-1,1,-1,1,-1},
                  {1,2,3,4,5,6},
                  {-6,-5,-4,-3,-2,-1},
              };
    cout<<fun(a, 6);
    return 0;
}
```

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án:	Thang điểm	Ghi chú
1	0.5đ	Bất kỳ 1 số nào khác đều là sai, 0đ

Câu 13: (1 điểm) Cho một chương trình như sau. Hãy cho biết dòng lệnh nào sai (sai về cú pháp hoặc logic), và cho biết tại sao sai?

STT	Chương trình	Giải thích tại sao các dòng lệnh bị sai
1.	#include <iostream>	Dòng 7: d++;
2.	using namespace std;	-Tên mảng d là một hằng con trả nên không thể
3.	int main()	thay đổi giá trị của hằng này

<pre> 4. { 5. int d[]={1,2,3,4,5}; 6. * (d+3)=2; 7. d++; 8. cout<<d; 9. int a=5,b=6,c,*pa=NULL; 10. double *pb,*pc; 11. &pb=0x6efef1f; 12. pc=&c; 13. *pa = &a; 14. return 0; 15. }</pre>	<p>Dòng 11: &pb=0x6efef1f;</p> <p>-Không thể thay đổi địa chỉ của một biến trong bộ nhớ (không được quyết định bởi người sử dụng)</p> <p>- Hoặc, sai cú pháp của phép gán, về phải của phép gán phải là biến.</p> <p>Dòng 12: pc=&c;</p> <p>-Không tương thích về kiểu dữ liệu</p> <p>Dòng 13: *pa = &a;</p> <p>-Sai cú pháp gán, đúng phải là pa=&a</p>
--	---

Đáp án:	Thang điểm	Ghi chú
Có 4 dòng lệnh sai (báo lỗi)	Chỉ ra đúng 1 dòng lệnh bị sai/lỗi và giải thích “chấp nhận được” sẽ đạt 0.25đ	<p>-Chỉ khi nào giải thích thuyết phục mới có điểm, dù xác định được dòng lệnh sai nhưng giải thích không đạt vẫn không có phần điểm nào</p> <p>-Câu lệnh không sai/lỗi nhưng SV cho rằng sai thì “BỊ ĐÁNH DẤU TRỪ 0.125”</p> <p>-Nếu chỉ có 1 lần “BỊ ĐÁNH DẤU TRỪ 0.125” thì bỏ qua KHÔNG TRỪ</p> <p>- Nếu nhiều hơn 1 lần “BỊ ĐÁNH DẤU TRỪ 0.125” thì PHẢI TRỪ theo đúng số lần bị đánh dấu</p> <p>- Câu cout<<d; không bị lỗi vì chương trình có thể hiểu là xuất địa chỉ của mảng d, nhưng nếu SV cho rằng “lỗi xuất mảng, phải dùng vòng lặp để xuất từng phần tử mảng” thì KHÔNG tính điểm cũng không trừ điểm.</p>

Ví dụ (một số câu trả lời của SV đã được đánh giá như sau)	ĐẠT 0.25đ	KHÔNG THUYẾT PHỤC, KHÔNG CÓ ĐIỂM
Dòng 7: d++;	(châm chước những câu trả lời sau) - “d là mảng, không thể thực hiện phép toán” - “Không thể cộng mảng” - “d là mảng, không phải giá trị” - “thiếu xác định vị trí do giá trị của d là con trỏ, không phải số”	- “chưa cấp phát bộ nhớ cho d”(SAI) - “sai vì không chỉ rõ tăng giá trị ở phần tử nào” (không thuyết phục)
Dòng 11: &pb=0x6efe1f;	- “không thể thay đổi được địa chỉ của pb” - “sai vì không thể tự cấp địa chỉ của 1 biến” - “hệ thống tự cấp phát địa chỉ, không gán được” - “do địa chỉ là duy nhất và do chương trình cấp khi khai báo double *pb,*pc;ở dòng 10”	- “pb là con trỏ, không thể gán giá trị cho pb bằng lệnh này” (không thuyết phục) - “pb đã trỏ tới địa chỉ của biến nên không thể thêm & nữa” (không thuyết phục)
Dòng 12: pc=&c;	- “Không cùng kiểu dữ liệu giữa con trỏ pc và c” - “c kiểu int còn pc kiểu double” - “sai do con trỏ kiểu double không thể trỏ đến biến kiểu int”	- “sai cú pháp, sửa lại là &pc=c mới đúng” (SV còn sai nghiêm trọng hơn)
Dòng 13: *pa = &a;	- “*pa là giá trị của vùng nhớ mà con trỏ pa trỏ tới, có kiểu dữ liệu là int, nên không được gán vào vùng nhớ đó 1 địa chỉ (&a)” - “*pa là giá trị của biến mà pa trỏ tới nên không thể bằng địa chỉ của a”	
Dòng 5: int d[]={1,2,3,4,5};	KHÔNG SAI nhưng SV trả lời là: “sai vì phải khai báo kích thước mảng, cú pháp đúng phải là int d[5] = ” → ĐÁNH DẤU TRÙ 0.125	
Dòng 6: *(d+3)=2;	KHÔNG SAI nhưng SV trả lời là: “sai vì giá trị trong mảng d không thể thay đổi được”, “mảng là con trỏ nên không thể thay đổi địa chỉ được”, “*(d+3) sai cú pháp”, “d là mảng chứ không phải là biến”, “chưa khai báo con trỏ” → ĐÁNH DẤU TRÙ 0.125	
Dòng 8: cout<<d;	KHÔNG SAI nhưng SV trả lời là “d là mảng, không thể	

	xuất mảng như vậy, chỉ có thể xuất từng phần tử của mảng dùng for() cout<<d[i]”, “sai logic vì khi đó sẽ in ra địa chỉ” → KHÔNG tính điểm cũng không trừ điểm.
Dòng 9: int a=5,b=6,c,*pa=NULL;	KHÔNG SAI nhưng SV trả lời là: “lỗi do không được khai báo con trỏ chung với biến kiểu int” → ĐÁNH DẤU TRỪ 0.125
Dòng 10: double *pb,*pc;	KHÔNG SAI nhưng SV trả lời là: “lỗi do không cung cấp địa chỉ cho con trỏ pb, pc”, “sai do chưa khởi tạo giá trị cho con trỏ pb,pc”, “sai do con trỏ chưa trỏ vào đâu”, “sai do con trỏ không phải là kiểu số thực” → ĐÁNH DẤU TRỪ 0.125

Câu 14: (0.75 điểm) Cho một đoạn chương trình như sau, hãy giải thích ý nghĩa của từng câu lệnh trong hàm main từ dòng số 5 đến dòng số 8 và cho biết kết quả in ra màn hình sau khi thực hiện xong chương trình.

Dòng lệnh số	Chương trình	Giải thích ý nghĩa của từng câu lệnh trong hàm main từ dòng số 5 đến dòng số 8
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14.	#include <iostream> using namespace std; int main(){ int a[5]={5,4,3,2,1}; int *p=&a[2]; *p=7; p[-1]=0; p++; *(p+1)=-5; p=new int(10); for (int i=0;i<5;i++) cout<<a[i]<<" "; return 0; }	<p>Dòng 5: khai báo con trỏ p kiểu int, và khởi tạo giá trị của p là địa chỉ của a[2]</p> <p>Hoặc chỉ cần chú trọng thông tin quan trọng nhất: p trỏ tới a[2], hoặc p chứa địa chỉ của a[2]</p> <p>Dòng 6: cập nhật giá trị của vùng nhớ mà p quản lý là 7 hay cập nhật a[2]=7</p> <p>Dòng 7: a[1]=0</p> <p>Dòng 8: p trỏ tới a[3], hoặc p chứa địa chỉ của a[3]</p> <p>//Lưu ý: còn 1 câu hỏi khác ở bên dưới</p>

Đáp án:	Thang điểm	Ghi chú
Yêu cầu SV giải thích 4 dòng lệnh như trên	Chỉ cần giải thích thuyết phục 1 dòng lệnh nào đó là đạt 0.25đ Nên chỉ cần 2 dòng chấp nhận	Vì đã chấm rộng rãi như vậy nên đòi hỏi SV phải giải thích chính xác, dùng từ ngữ đúng, không mơ hồ, không rối ren

	được là đạt trọn 0.5đ và chỉ tối đa 0.5đ cho phần giải thích này	gây khó hiểu cho người đọc Nếu người đọc phải tốn công sức và thời gian để hiểu câu trả lời của SV thì xem như không đạt.
--	---	--

Ví dụ (một số câu trả lời của SV đã được đánh giá như sau)	ĐẠT 0.25đ	KHÔNG THUYẾT PHỤC, KHÔNG CÓ ĐIỂM
Dòng 5: int *p=&a[2];	<ul style="list-style-type: none"> - “con trả p trả tới địa chỉ vùng nhớ của a[2], nơi lưu giá trị của a[2]” - “p lấy bằng địa chỉ của phần tử thứ a[2] trong mảng a” - “p trả đến vị trí thứ 3 của mảng a” (châm chước được) - “p trả đến phần tử thứ 3 của mảng a” - “p trả tới địa chỉ của ô thứ 3 trong mảng a” (châm chước được) - “gán cho p địa chỉ của phần tử có index là 2 trong mảng” 	<ul style="list-style-type: none"> -“thay đổi vùng trả p thành vị trí thứ 2 trong mảng a” (SAI vì vị trí thứ 3 hoặc index 2 mới đúng) -“gán con trả p vào vị trí phần tử số 2 của mảng a” (người đọc không hiểu) -“p bằng với địa chỉ a[2] nên p=a[2]” (câu trả lời không chấp nhận được) -“gán giá trị a[2] vào p” (gán địa chỉ chứ sao là giá trị) -“gán giá trị con trả p (tức địa chỉ) vào địa chỉ của a[2]” (GV đọc ko hiểu) -“a[2] là tham số chiếu lên p, a[2] thay đổi thì p thay đổi theo” -“gán địa chỉ của con trả p vào địa chỉ của a[2]” (dùng từ sai)
Dòng 6: *p = 7;	<p>(thiếu thông tin nhưng châm chước được)</p> <ul style="list-style-type: none"> - “gán cho giá trị tại địa chỉ mà p trả tới =7” - “giá trị con trả p trả vào bằng 7” - “gán cho địa chỉ con trả p trả vào bằng 7” - “giá trị của vùng nhớ mà p trả tới bằng 7” - “gán giá trị cho nơi mà p trả tới là 7” - “gán giá trị mà p trả tới (a[2]) bằng 7” 	<ul style="list-style-type: none"> -“gán giá trị đầu của mảng a[0]=7” (SAI) -“trả con trả p tới vị trí bằng 7” (GV đọc ko hiểu) -“gán giá trị 7 cho *p” (đọc nội dung câu lệnh chứ không phải giải thích ý nghĩa) -“thay đổi giá trị của con trả p thành 7” (SAI) -“gán giá trị cho con trả p bằng 7” (SAI)

	<ul style="list-style-type: none"> -“p trả về số 7” -“gán giá trị 7 cho biến mà p đang trả tới” 	
Dòng 7: p[-1]=0;	<ul style="list-style-type: none"> -“phần tử thứ 2 của mảng a có giá trị là 0” -“gán giá trị của phần tử trước phần tử p trả tới bằng 0” 	<ul style="list-style-type: none"> -“cho phần tử thứ p[-1] =0” (không có thông tin gì mới) -“con trỏ p lùi về 1 vị trí và gán giá trị được lưu ở vị trí đó bằng 0” (SAI vì p không thay đổi) -“gán a[-1]=0” (SAI vì a[1] mới đúng, a[-1] truy cập index âm) -“p lùi xuống 1 đơn vị nên p=a[1] và cho nó giá trị là 0” (câu trả lời không chấp nhận được) -“con trỏ p[-1] trả tới a[1] và gán cho a[1] giá trị mới là 0” (con trỏ p[-1] là con trỏ gì, ở đâu mà có)
Dòng 8: p++;	<ul style="list-style-type: none"> -“dời con trỏ đến vị trí thứ 4 của mảng a” -“p trả tới vị trí đứng sau , so với vị trí hiện tại đang trả” -“cho p trả vào phần tử kế tiếp của mảng a” 	<ul style="list-style-type: none"> -“p trả tới vị trí a[2]” (SAI vì a[3] mới đúng) -“p tăng lên 1 giá trị tức p=p+1” (không có thông tin gì mới) -“cộng thêm cho con trỏ p 1 giá trị” (người đọc không hiểu) -“đổi giá trị của vùng p+0 thành giá trị của vùng p+1, tức a[2]=a[3]” (SAI) -“tăng giá trị cho p” (không rõ) - “tăng giá trị của a[2] lên 1 đơn vị” (SAI vì câu lệnh cập nhật con trỏ) - “tăng địa chỉ a[2] lên a[3]” (GV không hiểu) -“tăng p lên 1 đơn vị tức *p từ 7 thành 8” (SAI)

Sau khi thực hiện xong chương trình, kết quả in ra màn hình là:

Đáp án:	Thang điểm	Ghi chú
5 0 7 2 -5	0.25đ	<p style="color: red;">Đúng hết dãy số mới có điểm, sai 1 chữ số cũng không có điểm nào</p> <p>Viết các số không cách nhau bởi khoảng trắng cũng không trừ điểm</p>

Câu 15: (0.75 điểm) Điền vào các vị trí trống để hoàn chỉnh một chương trình cho phép nhập một đa giác (DAGIAC) có n điểm (DIEM), với $3 \leq n \leq 10$. Chương trình yêu cầu người dùng nhập số điểm của đa giác, sau đó nhập tọa độ của từng điểm. Ví dụ, khi nhập đa giác ABCD, cần phải nhập tọa độ của 4 điểm (A, B, C, D) tạo thành đa giác. Chú ý, không cần kiểm tra tính hợp lệ của đa giác và phải tận dụng được hàm nhập một điểm (NhậpMotDiem) trong hàm nhập một đa giác (NhậpMotDaGiac).

```
#include<iostream>
using namespace std;
#define MAXN 10
struct DIEM {
    int x, y;
};
struct DAGIAC{
    int n;
    DIEM *p;
};

void NhậpMotDiem(DIEM &a) {
    cin>>a.x>>a.y;
}
```

(1)

(2)

Vị trí	Đáp án	Đáp án khác	Thang điểm	Ghi chú:
(1) (2)	DIEM &a cin>>a.x>>a.y;	DIEM *a cin>>a->x>>a->y; hoặc cin>>(*a).x>>(*a).y;	0.25đ	-Đúng hết cả 2 mới đạt 0.25đ, sai hoặc thiếu 1 trong 2 là 0đ
Những trường hợp sau là sai, 0đ	Thiếu dấu & là sai, 0đ, không chia điểm nhỏ hơn nữa	Vì a là con trỏ nên khi truy cập thuộc tính phải dùng a->x hoặc (*a).x . Nếu SV dùng lỗn lộn toán tử là sai, 0đ		

```

(1)                                     (2)
DAGIAC NhapMotDaGiac (-----) {
    DAGIAC d;   (3)
    cin>>d.n;   (4)
    d.p = new DIEM[d.n];   (6)
    for(int i=0;i<d.n;i++)
        NhapMotDiem(d.p[i]);   (7)
    return d;   (5)
}

```

Đáp án	Thang điểm	Ghi chú:
Dúng (1,2,3,4,5)	0.25đ	Nếu đúng 5 vị trí này thì đã đạt 0.25đ, tức SV sẽ đạt tối thiểu 0.25đ cho phần NhapMotDaGiac
Dúng (6,7)	0.25đ	
Một số trường hợp đặc biệt:		
<p>1. Nếu thiếu (hoặc sai) vị trí (6), tức không cấp phát vùng nhớ (hoặc cấp phát sai, VD <code>d.p = new int[d.n]</code> , <code>DAGIAC *a = new DIEM[a->n]</code> là SAI) cho mảng các DIEM thì MẤT 0.25đ</p> <ul style="list-style-type: none"> + chỉ sai thêm 1 lỗi khác (các vị trí còn lại đúng) thì bỏ qua, có thể đạt 0.25đ cho phần NhapMotDaGiac + sai nhiều hơn 1 lỗi khác thì 0đ cho phần NhapMotDaGiac <p>2. Nếu (6) đúng,</p> <ul style="list-style-type: none"> + chỉ sai 1 hoặc 2 lỗi ở những vị trí còn lại thì chấm chước được, có thể đạt 0.5đ cho phần NhapMotDaGiac + sai nhiều hơn 2 lỗi thì 0đ cho phần NhapMotDaGiac <p>3. Nếu hàm NhapMotDiem định nghĩa như sau: <code>void NhapMotDiem(DIEM *a){ cin>>a->x>>a->y; }</code> thì lời gọi hàm phải truyền địa chỉ mới đúng, tức <code>NhapMotDiem(&d.p[i])</code>, thiếu & là sai, MẤT 0.25đ dù (6) đúng</p> <p>4. Nếu hàm NhapMotDiem định nghĩa như sau: <code>void NhapMotDiem(int a, int b){ cin>>a>>b; }</code>, không có điểm, tức 0đ cho hàm NhapMotDiem</p>		

```

int main() {
    DAGIAC B;
    B = NhapMotDaGiac();
    //Xử lý đa giác
    return 0;
}

```

Câu 16: (0.75 điểm) Cho một bài toán như sau:

Input: Một số nguyên dương n với $1 \leq n \leq 1$ tỷ.

Output: Số lượng hình chữ nhật có diện tích nhỏ hơn n mà cả chiều dài, chiều rộng và đường chéo đều là số nguyên dương? (Một hình chữ nhật có thể xoay ngang hay dọc tùy ý, tức là hình chữ nhật với kích thước 4×3 cũng chính là hình chữ nhật với kích thước 3×4 , chúng ta chỉ đếm một hình)

a. Cho một ví dụ với bài toán trên. Giải thích vì sao có kết quả như vậy? (tức là cho biết chiều dài, chiều rộng, đường chéo và diện tích của các hình chữ nhật thỏa điều kiện)

Input:				
1 số n bất kỳ thuộc khoảng	$12 < n < 49$	$48 < n < 61$	$60 < n < 109$	$1 \leq n \leq 12$
Ví dụ	13	49	61	1
Output:	1	2	3	0
Các hình chữ nhật thỏa điều kiện là:	$(3,4)/(4,3)$ $(6,8)/(8,6)$	$(3,4)/(4,3)$ $(6,8)/(8,6)$	$(3,4)/(4,3)$ $(6,8)/(8,6)$ $(5,12)/(12,5)$	1 số SV chọn trường hợp này vẫn có điểm (nếu có giải thích)

Giải thích: Hình chữ nhật $(3, 4)$ với chiều rộng là 3, chiều dài là 4, đường chéo $\sqrt{3^2 + 4^2} = 5$, đều là các số nguyên dương. Diện tích $3 \times 4 = 12 < n = 13$

b. Tìm một input mà có output là 3 và giải thích vì sao có kết quả như vậy?

Input:	Ghi chú
1 số n bất kỳ thuộc khoảng	n SAI thì không chấm phần giải thích, 0đ.
Ví dụ	61
Output:	3
3 hình chữ nhật bao gồm:	Tuy nhiên, sẽ có chấm chước cho tình huống như bên dưới

Giải thích:

STT	Chiều dài	Chiều rộng	Đường chéo	Diện tích
1	3	4	5	12
2	6	8	10	48
3	5	12	13	60

Nhiều SV chọn n = 109 hay n = 110 và khi giải thích chỉ ra HCN thứ 3 là (9, 12) chứ không phải là (5, 12) như đáp án thì là SAI, nhưng sẽ chấm chước, đạt 0.25đ cho câu b

- Chỉ có duy nhất 1 trường hợp chấm chước đạt 0.25đ, đó là khi SV liệt kê 3 HCN $\{(3,4), (6,8), (9,12)\}$
- Bất kỳ bộ 3 HCN nào khác đều 0đ: Ví dụ $\{(3,4), (6,8), (12, 16)\}$, $\{(3,4), (5,12), (9, 12)\}$, $\{(3,4), (5,12), (12, 16)\}$,--> 0đ

3. Nếu cả câu a và b SV đều chọn {(3,4),(6,8), (9,12)} thì đạt 0.5đ cho câu 16

	9	12	15	108 (Đây là đáp án SAI, vì output khi đó là 4 chứ không phải 3)
--	---	----	----	---

HẾT