

**ĐỀ 3**

**MÃ ĐỀ: Tôi trung thực**

**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<<17.0/4-24%3<<", ";
    cout<<(int)5.12*(20/6)+(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
4.25	16	<b>Đúng 1 số đạt 0.25đ</b>	-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) || x-y<=x)<<", ";
    cout<<(x==y/2?-x:y-x)<<", ";
    cout<<(x<y<2)<<", ";
    cout<< (!2 && !(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
1	-12	1	0	<b>Đúng 1 số đạt 0.25đ</b>	<ul style="list-style-type: none"><li>- Nhiều SV ghi đáp án là “ , -12, ,0 ”, hoặc “1,-12,” vẫn chấm theo vị trí của các số</li><li>- Nhiều SV chỉ ghi 2-3 số mà không đủ 3 dấu , phân cách (VD “1,-12” ) thì 0đ</li></ul>

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

```
#include <iostream>
using namespace std;
int main()
{    int x,y;
    cin >> x;
    y=2*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 **4** thì kết quả in ra màn hình là:

Đáp án: có 2 số	Thang điểm
-8	-2

**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 < 25);
    cout<<"sum="<<sum;
    return 0;
}
```

Giả sử user muốn nhập lần lượt các số sau **10 1 2 30 9 10 15 -9 50 7 11 42** (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=31	0.5đ
<b>31(Chỉ ghi duy nhất số 31)</b>	<b>0.25đ</b>

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

```
#include <iostream>
using namespace std;
int main ()
{
    int n=318, 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
352	<b>0.5đ</b>	352 là 1 số. Bất kỳ 1 số nào khác đều là sai, 0đ. VD, nhiều SV trả lời là 353, 354, 35 thì là sai, tức 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(y, z, x);
    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
10	5	5	<b>-Chỉ sai 1 số: 0.25đ -Sai 2 số: 0đ</b>	Chỉ có 3 mức điểm:0, 0.25, 0.5, không chia nhỏ hơn nữa SV ghi 10,5,5 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--;
    return a;
}
int main()
{
    int a=7;
    int &b=a;
    cout<<test(a)<<", ";
    b=100;
    a=b-a;
    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
6	1	1	<b>-Chỉ sai 1 số: 0.25đ -Sai 2 số: 0đ</b>	-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(2)+fun(4);
    return 0;
}
```

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

Đáp án:	Thang điểm	Ghi chú
63	<b>0.5đ</b>	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=0; i<n-2; i++)
        if (a[i] > a[i+1]&& a[i+1]<a[i+2]) return true;
        return false;
}
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 1	<b>0</b> <b>false</b>	<b>0.25đ</b>	Chấm theo cặp. Ở mỗi cặp, phải đúng hết cả 2 mới đạt 0.25 điểm, sai 1 số là 0đ
2 5 3	<b>0</b> <b>false</b>		
5 3 3 6 4 5	<b>1</b> <b>true</b>	<b>0.25đ</b>	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	<b>0</b> <b>false</b>		

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

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

{    int i=0, dem=0;

    while(s[i]!='\0')      (1)
    {
        if('0'<=s[i]&&s[i]<='9') (2)
            dem++;

        i++;                  (3)
    }

    return dem;
}

int main()
{
    char str[35];
    cin.getline(str, 30);
    cout<< count_Digit(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	<b>SAI: KHÔNG CÓ ĐIỂM</b>  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)	'0'<=s[i]&&s[i]<='9'	'0'<=s[i] and s[i]<='9' 48<=s[i]&&s[i]<=57 '0'<=*(s+i)&&*(s+i)<='9'	0.25đ	<b>SAI: KHÔNG CÓ ĐIỂM</b>  '0'<=s[i] (chỉ so với 0 hoặc chỉ so với 9)

				'0'<=s[i]<='9' (không có &&)  "0"≤s[i] && s[i]≤"9" (dùng nháy kép, không phải ký tự)  0≤s[i]&&s[i]≤9 (thiếu dấu nháy đơn, không phải ký tự)  '0'≤s[i]    s[i]≤'9' (sai dấu   )  '0'<s[i]&&s[i]<'9' (thiếu =)  s[i]='0' to '9'  s[i]==0    s[i]==1    s[i]==2....(không so kí tự)
(3)	i++; return dem;	<b>0.25đ</b>		Đúng hết cả 2 mới đạt 0.25đ, sai/thiếu 1 trong 2 là 0đ
	<b>Nhiều SV dùng sai tên chuỗi, thay vì s[i] thì lại gọi a[i], str[i] thì TRỪ 0.25đ</b>			

**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},
                  {4,3,0,1,5},
                  {1,2,3,4,5}};

    int S=0;
    for(int i=1; i<4;i++)
        for(int j=2;j<5;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
17	<b>0.5đ</b>	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},
                 {-6,-5,-4,-3,-2,-1},
                 {1,-1,1,-1,1,-1},
                 {1,2,3,4,5,6}};
    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
3	<b>0.5đ</b>	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>	<b>Dòng 6: d++;</b>
2.	using namespace std;	<b>-Tên mảng d là một hằng con trỏ nên không thể thay đổi giá trị của hằng này</b>
3.	int main()	
4.	{	<b>Dòng 11: int *pc=&amp;c;</b>
5.	int d[]={1,2,3,4,5};	<b>-Không tương thích về kiểu dữ liệu</b>
6.	<b>d++;</b>	<b>Dòng 12: pb = &amp;(b+2);</b>
7.	cout<<d;	<b>-Không thể tham chiếu một con trỏ đến một</b>
8.	* (d+1)=8;	

9. 10. 11. 12. 13. 14. 15.	<pre> double a,b=6.3,c=0.1;  int *pa,*pb=NULL;  int *pc=&amp;c;  pb = &amp;(b+2);  &amp;a = pb;  return 0;  } </pre>	<p><b>hằng số hay một biểu thức</b></p> <p><b>-Hoặc, không thể lấy địa chỉ của một hằng số hay một biểu thức</b></p> <p><b>Dòng 13: &amp;a = pb;</b></p> <p><b>-Không thể thay đổi địa chỉ của một biến trong bộ nhớ (không được quyết định bởi người sử dụng)</b></p> <p><b>- Hoặc, sai cú pháp của phép gán, về phải của phép gán phải là biến.</b></p>
--	--	---

Đáp án:	Thang điểm	Ghi chú
Có 4 dòng lệnh sai (báo lỗi)	<b>Chỉ ra đúng 1 dòng lệnh bị sai/lỗi và giải thích “chấp nhận được” sẽ đạt 0.25đ</b>	<ul style="list-style-type: none"> <li>-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</li> <li>-Câu lệnh không sai/lỗi nhưng SV cho rằng sai thì “BỊ ĐÁNH DẤU TRỪ 0.125”</li> <li>-Nếu chỉ có 1 lần “BỊ ĐÁNH DẤU TRỪ 0.125” thì bỏ qua KHÔNG TRỪ</li> <li>- Nếu nhiều hơn 1 lần “BỊ ĐÁNH DẤU TRỪ 0.125” thì PHẢI TRỪ theo đúng số lần bị đánh dấu</li> <li>- Câu cout&lt;&lt;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.</li> </ul>

Ví dụ (một số câu trả lời của SV đã được đánh giá như sau)	<b>ĐẠT 0.25đ</b>	<b>KHÔNG THUYẾT PHỤC, KHÔNG CÓ ĐIỂM</b>
Dòng 6: d++;	<ul style="list-style-type: none"> <li>(châm chước những câu trả lời sau)</li> <li>-“d là mảng, không thể thực hiện phép toán”</li> <li>-“Không thể cộng mảng”</li> <li>-“d là mảng, không phải giá trị”</li> <li>-“thiếu xác định vị trí do giá trị của d là con trỏ, không phải số”</li> </ul>	<ul style="list-style-type: none"> <li>-“chưa cấp phát bộ nhớ cho d”(SAI)</li> <li>-“sai vì không chỉ rõ tăng giá trị ở phần tử nào” (không thuyết phục)</li> </ul>

Dòng 11: int *pc=&c;	<ul style="list-style-type: none"> <li>-“Không cùng kiểu dữ liệu giữa con trỏ pc và c”</li> <li>-“c kiểu double còn pc kiểu int”</li> <li>-“sai do con trỏ kiểu int không thể trỏ đến biến kiểu double”</li> </ul>	<ul style="list-style-type: none"> <li>-“sai cú pháp, sửa lại là &amp;pc=c mới đúng” (SV còn sai nghiêm trọng hơn)</li> </ul>
Dòng 12: pb = &(b+2);	<ul style="list-style-type: none"> <li>-“(b+2) không phải là 1 biến nên không có địa chỉ”</li> <li>-“không thể lấy địa chỉ của b+2”</li> </ul>	<ul style="list-style-type: none"> <li>-“sai cú pháp, pb = &amp;b mới đúng, b truyền tham chiếu cho pb”</li> <li>-“sai vì b không phải con trỏ”</li> <li>-“thiếu int*, đúng phải là int* pb=&amp;(b+2)”</li> <li>-“địa chỉ không chứa biểu thức” (đọc không hiểu)</li> <li>-“sai vì b là biến double, không phải mảng và pb là con trỏ int, khác double”</li> <li>-“vì b là 1 biến chứ không phải là mảng nên khi con trỏ pb trỏ tới &amp;(b+2) thì con trỏ này ko trỏ đến đâu cả”</li> </ul>
Dòng 13: &a = pb;	<ul style="list-style-type: none"> <li>-“không thể thay đổi được địa chỉ của a”</li> <li>-“sai vì không thể tự cấp địa chỉ của 1 biến”</li> <li>-“hệ thống tự cấp phát địa chỉ, không gán được”</li> <li>-“do địa chỉ là duy nhất và do chương trình cấp khi khai báo double a ở dòng 9”</li> <li>-“&amp;a là địa chỉ, pb là con trỏ, không thể gán con trỏ cho địa chỉ mà chỉ có thể ngược lại”</li> </ul>	<ul style="list-style-type: none"> <li>-“Không cùng kiểu dữ liệu giữa a và con trỏ pb”</li> <li>-“sai, đúng phải là pb=&amp;a”</li> <li>-“sai cho chưa có pb”, “sai do pb chưa trỏ tới ai”</li> <li>-“sai vì pb là con trỏ”</li> <li>-“sai vì a là biến double, pb là con trỏ kiểu int, không thể chia sẻ cùng 1 địa chỉ”</li> </ul>
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] = ” → <b>ĐÁNH DẤU TRÙ 0.125</b>	
Dòng 8: *(d+1)=8;	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+1) sai cú pháp”, “d là mảng chứ không phải là biến”, “chưa khai báo con trỏ” → <b>ĐÁNH DẤU TRÙ 0.125</b>	

Dòng 10: int *pa,*pb=NULL;	KHÔNG SAI nhưng SV trả lời là: “lỗi do không cung cấp địa chỉ cho con trỏ pa”, “sai do chưa khởi tạo giá trị cho con trỏ pa”, “sai do con trỏ pa chưa trỏ vào đâu” → <b>ĐÁNH ĐẤU TRÙ 0.125</b>
Dòng 7: 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” → <b>KHÔNG tính điểm cũng không trừ điểm.</b>

**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[]={5,4,3,2,1};  int *p=new int(10);  *p = 0;  p=&a[3];  *(p-1)=1;  p--;  p[2]=5;  for(int i=0;i<5;i++)  cout<<a[i];  return 0; }	<p><b>Dòng 5:</b> tạo một biến động kiểu int, cho con trỏ p quản lý (hay trỏ tới) biến động mới, khởi tạo giá trị của biến là 10</p> <p><b>Dòng 6:</b> cập nhật giá trị của vùng nhớ mà p quản lý là 0 (hay cập nhật giá trị của biến động vừa tạo ở trên là 0)</p> <p><b>Dòng 7:</b> p trỏ tới (hay chưa địa chỉ của) a[3] (hoặc phần tử thứ 4 của mảng)</p> <p><b>Dòng 8:</b> a[2]=1</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	<b>Chỉ cần giải thích thuyết phục 1 dòng lệnh nào đó là đạt 0.25đ</b>  <b>Nên chỉ cần 2 dòng chấp nhận được là đạt trọn 0.5đ và chỉ tối đa 0.5đ cho phần giải thích này</b>	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 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)	<b>DẠT 0.25đ</b>	<b>KHÔNG THUYẾT PHỤC, KHÔNG CÓ ĐIỂM</b>
Dòng 5: int *p=new int(10);	<ul style="list-style-type: none"> <li>-“tạo 1 vùng nhớ, cho con trỏ p trỏ tới vùng nhớ mang giá trị 10”</li> <li>“cấp phát động vùng nhớ kiểu int mang giá trị 10 cho p”</li> <li>“tạo con trỏ p, trỏ tới một vùng nhớ mới có giá trị là 10”</li> </ul>	<ul style="list-style-type: none"> <li>-“tạo con trỏ p kiểu int có giá trị 10” (SAI)</li> <li>“tạo mảng kiểu int có 10 phần tử”(SAI)</li> <li>“khởi tạo mảng động gồm 10 giá trị” (SAI)</li> <li>“khởi tạo vùng nhớ int với giá trị 10 cho con trỏ p” (GV đọc thấy khó hiểu)</li> <li>“khởi tạo con trỏ p trỏ tới vùng có giá trị 10” (thiếu tạo biến động nên chưa thuyết phục lắm)</li> <li>“cấp phát động cho con trỏ p” (chưa đủ thông tin)</li> <li>“cấp phát động cho con trỏ p và con trỏ p có giá trị là 10”</li> <li>“gán giá trị cho *p là 10” (chưa đủ thông tin)</li> <li>“cấp phát bộ nhớ động với kiểu số nguyên bằng 10 cho con trỏ p”</li> </ul>
Dòng 6: *p = 0;	<ul style="list-style-type: none"> <li>-“gán cho giá trị tại địa chỉ mà p trỏ tới =0”</li> <li>“giá trị con trỏ p trỏ vào bằng 0”</li> <li>“gán cho địa chỉ con trỏ p trỏ vào bằng 0”</li> <li>“giá trị của vùng nhớ mà p trỏ tới bằng 0”</li> <li>“gán giá trị cho nơi mà p trỏ tới là 0”</li> <li>“gán giá trị mà p trỏ tới bằng 0”</li> <li>“p trỏ về số 0”</li> <li>“gán giá trị 0 cho biến mà p đang trỏ tới”</li> </ul>	<ul style="list-style-type: none"> <li>-“gán giá trị đầu của mảng p[0]=0” (SAI)</li> <li>“trỏ con trỏ p tới vị trí bằng 0” (GV đọc ko hiểu)</li> <li>“gán giá trị 0 cho *p” (đọc nội dung câu lệnh chử không phải giải thích ý nghĩa)</li> <li>“thay đổi giá trị của con trỏ p thành 0” (SAI)</li> <li>“gán giá trị cho con trỏ p bằng 0” (SAI)</li> </ul>
Dòng 7: p=&a[3];	<ul style="list-style-type: none"> <li>-“con trỏ p giờ sẽ trỏ tới địa chỉ của phần tử thứ 4 của mảng a”</li> <li>-“con trỏ p trỏ đến địa chỉ</li> </ul>	-“thay đổi vùng trỏ p thành vị trí thứ 3 trong mảng a” (SAI vì vị trí thứ 4 hoặc index 3 mới đúng)

	a[3]"  - “gán giá trị của con trỏ p là địa chỉ của phần tử a[3]”  - “p không trỏ đến vùng nhớ lúc nãy nữa, p trỏ đến phần tử thứ 4 của mảng a” - “p trỏ đến vị trí thứ 4 của mảng a” (châm chước được) - “p trỏ đến phần tử thứ 4 của mảng a” - “p trỏ tới địa chỉ của ô thứ 4 trong mảng a” (châm chước được) - “gán cho p địa chỉ của phần tử có index là 3 trong mảng”	- “gán giá trị a[3] vào p” (gán địa chỉ chứ sao là giá trị) - “gán giá trị con trỏ p (tức địa chỉ) vào địa chỉ của a[3]” (GV đọc ko hiểu) - “a[3] là tham số chiếu lên p, a[3] thay đổi thì p thay đổi theo” - “gán địa chỉ của con trỏ p vào địa chỉ của a[3]” (dùng từ sai) - “gán con trỏ p vào vị trí phần tử số 3 của mảng a” (người đọc không hiểu) - “p bằng với địa chỉ a[3] nên p=a[3]” (câu trả lời không chấp nhận được)
Dòng 8: *(p-1)=1;	- “phần tử thứ 3 của mảng a có giá trị là 1” - “gán giá trị của phần tử trước phần tử p trỏ tới bằng 1”	- “gán 1 vào con trỏ p-1” (GV đọc ko hiểu) “gán giá trị của *(p-1) = 1” (đọc nội dung câu lệnh chứ không phải giải thích ý nghĩa)  - “con trỏ p lùi về 1 vị trí và gán giá trị được lưu ở vị trí đó bằng 1” (SAI vì p không thay đổi) - “p lùi xuống 1 đơn vị nên p =a[2] và cho nó giá trị là 1” (câu trả lời không chấp nhận được)

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

Đáp án:	Thang điểm	Ghi chú
54125	<b>0.25đ</b>	<b>Đúng hết dãy số mới có điểm, sai 1 chữ số cũng không có điểm nào</b>

**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 NhaphMotDiem(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	<b>Thiếu dấu &amp; là sai, 0đ</b> , không chia điểm nhỏ hơn nữa	Vì a là con trỏ nên khi truy cập thuộc tính phải dùng a->x hoặc (*a).x . <b>Nếu SV dùng lỗn lộn toán tử là sai, 0đ</b>		

(1)

(2)

**DAGIAC** NhaphMotDaGiac (-----) {

**DAGIAC d;**

(3)

**cin>>d.n;**

(4)

**d.p = new DIEM[d.n];**

(6)

**for(int i=0;i<d.n;i++)**

(7)

**NhaphMotDiem(d.p[i]);**

**return d;**

(5)

}

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

```
int main() {
    DAGIAC B;
    B = NhậpMotDiem();
    //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  $4 \times 3$  cũng chính là hình chữ nhật với kích thước  $3 \times 4$ , 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)

<b>Input:</b>				
<b>1 số n bất kỳ thuộc khoảng</b>	<b>12&lt; n &lt;49</b>	<b>48&lt; n &lt;61</b>	<b>60&lt; n &lt;109</b>	<b>1&lt;= n &lt;12</b>

<b>Ví dụ</b>	<b>13</b>	<b>49</b>	<b>61</b>	<b>1</b>
<b>Output:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>0</b>
Các hình chữ nhật thỏa điều kiện là:	(3,4)/(4,3)	(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  $3*4=12 < n=13$

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	60< n <109	n SAI thì không chấm phần giải thích, 0đ.
<b>Ví dụ</b>	<b>61</b>	
Output:	3	
3 hình chữ nhật bao gồm:	(3,4)/(4,3) (6,8)/(8,6) (5,12)/(12,5)	Tuy nhiên, sẽ có chấm churorc 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 churorc, đạt 0.25đ cho câu b

- Chỉ có duy nhất 1 trường hợp chấm churorc 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
- 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