

## 2017 年秋：计算机高级语言程序设计(C++) 答案：

### 一、单项选择题（每题 2 分, 共 30 分）

- |        |        |        |        |        |
|--------|--------|--------|--------|--------|
| (1) C  | (2) C  | (3) B  | (4) C  | (5) C  |
| (6) C  | (7) A  | (8) C  | (9) D  | (10) B |
| (11) C | (12) D | (13) C | (14) A | (15) D |

### 二、程序阅读题（每题 3 分, 共 30 分）

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| (1) B | (2) C | (3) A | (4) D | (5) B  |
| (6) C | (7) C | (8) A | (9) D | (10) A |

### 三、编程题

(1)

```
#include <stdio.h>
#define N 4
#define M 5          /* 数组为 4 行 5 列 */
int main()
{
    int i,j,k,a[N][M],max,maxj,flag;
    printf("please input matrix:\n");
    for (i=0;i<N;i++)      /* 输入数组 */
        for (j=0;j<M;j++)
            scanf("%d",&a[i][j]);
    for (i=0;i<N;i++)
    {
        max=a[i][0];      /* 开始时假设 a[i][0]最大 */
        maxj=0;           /* 将列号 0 赋给 maxj 保存 */
        for (j=0;j<M;j++) /* 找出第 i 行中的最大数 */
            if (a[i][j]>max)
            {
                max=a[i][j]; /* 将本行的最大数存放在 max 中 */
                maxj=j;      /* 将最大数所在的列号存放在 maxj 中 */
            }
        flag=1;           /* 先假设是鞍点,以 flag 为 1 代表 */
        for (k=0;k<N;k++)
            if (max>a[k][maxj]) /* 将最大数和其同列元素相比 */
            {
                flag=0;      /* 如果 max 不是同列最小,表示不是鞍点令 flag1 为 0 */
                continue;
            }
        if(flag)           /* 如果 flag1 为 1 表示是鞍点 */
            {printf("a[%d][%d]=%d\n",i,maxj,max); /* 输出鞍点的值和所在行列号 */}
```

```

        break;
    }
}
if(!flag)                /* 如果 flag 为 0 表示鞍点不存在 */
    printf("It is not exist!\n");
return 0;
}

```

(2)

```

#include <iostream>
#include <string>
using namespace std;
template <typename Groap>Groap max(const Groap *r_array, int size) {
//这里 const 所放位置表示指针所指的对象不可改变, 可防止误改了实参
    Groap max_val=r_array[0];
    for (int i=1;i<size; ++i)
        if(r_array[i]>max_val) max_val=r_array[i];
    return max_val;
}
int ia[7]={10, 7, 14, 3, 25, 99, 100};
double da[6]={10. 2, 7. 1, 14. 5, 3. 2, 25. 6, 16. 8};
string sa[5]={"Shanghai", "Beijing", "Wuhan", "Daqing", "Xian"};
int main() {
    int i=max<int>(ia, 7);
    cout <<"整数最大值为: "<<i<<endl;
    double d=max(da, 6);
    cout <<"实数最大值为: "<<d<<endl;
    string s=max(sa, 5);
    cout <<"字典排序最大为: "<<s<<endl;
    return 0;
}

```

(3) 主函数:

```
#include <iostream>
#include "triangle.h"
using namespace std;
int main()
{
    int x1,y1;
    int x2,y2;
    int x3,y3;
    Triangle myTriAngle;
    bool bIsTri=false;
    while(1)
    {
        cout<<"Please input three Points' coordinate: "<<endl;
        cin>>x1>>y1>>x2>>y2>>x3>>y3;
        myTriAngle.AssignThreePoints(x1,y1,x2,y2,x3,y3);
        bIsTri=myTriAngle.bIsAriangle();
        if(bIsTri)
            break;
        else
            cout<<"Input is error!!!"<<endl;
    };
    cout<<"Perimeter of Triangle is:"<<myTriAngle.GetPeri()<<endl;
    cout<<"Area of Triangle is:"<<myTriAngle.GetArea()<<endl;
    return 0;
}

////////////////////////////////////
三角形类定义文件: triangle.h
class Triangle {
private:
    int x1,y1;  int x2,y2;  int x3,y3;
    double a,b,c;
    double dPeri, dArea;
public:
    Triangle(){};
    ~Triangle(){};
    void AssignThreePoints(int plx, int ply, int p2x, int p2y, int p3x, int p3y);
    bool bIsAriangle();
    void Triangle::GetEdgesLength();
    double GetPeri();
    double GetArea();
};

////////////////////////////////////
```

三角形类实现文件: triangle.cpp

```
#include <iostream>
#include "triangle.h"
#include <cmath>
using namespace std;

void Triangle::AssignThreePoints(int plx, int ply, int p2x, int p2y, int p3x, int p3y)
{
    x1=plx; y1=ply;
    x2=p2x; y2=p2y;
    x3=p3x; y3=p3y;
}

void Triangle::GetEdgesLength()
{
    a=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    b=sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));
    c=sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
}

bool Triangle::bIsAriangle()
{
    GetEdgesLength();
    if(a+b<=c || a+c<=b || b+c<=a)
        return false;
    else
        return true;
}

double Triangle::GetPeri()
{
    dPeri=a+b+c;
    return dPeri;
}

double Triangle::GetArea()
{
    double dTemp;
    dTemp=(a+b+c)/2;
    dArea=sqrt(dTemp*(dTemp-a)*(dTemp-b)*(dTemp-c));
    return dArea;
}
```