

類 科：資訊處理  
科 目：程式設計  
考試時間：2小時

更多高普考考試科目的歷屆題庫  
立即上網搜尋“高普考歷屆考古題懶人包”  
高鋒公職補習班整理  
Line@帳號：@gaofeng  
連絡電話：07-236-7296

座號：\_\_\_\_\_

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

## 一、關於以下C程式碼：

```
01 #include<stdio.h>
02 #define SIZE 10
03 #define THREE 3
04 unsigned int f1(unsigned int a, unsigned int b){ return (a&&!b); }
05 unsigned int f2(unsigned int a, unsigned int b){ return (a<<b); }
06 unsigned int f3(unsigned int a, unsigned int b){ return (a&~b); }
07 int f4(int a, int b) { return a*b+a-b; }
08 int f5(int a, int b) {
09     int data[SIZE];
10     for (int i=1, k=0; i<a; i++) {
11         if (i%3==0) data[k++]=f4(i, i+1);
12     }
13     return data[b];
14 }
15 int f6(int a, int b) {
16     int data[][THREE] = {{4,3,2},{3,4,2},{2,3,3}};
17     for (int i=0; i<THREE; i++) {
18         for (int j=0; j<THREE; j++) {
19             if (i>a || j>b)
20                 data[i][j]= data[j][i]+b;
21         }
22     }
23     return data[a][b];
24 }
25 int main() {
26     printf("%u\n", f1(6, 2));
27     printf("%u\n", f2(6, 2));
28     printf("%u\n", f3(7, 2));
29     printf("%d\n", f4(3, 12));
30     printf("%d\n", f5(15, 3));
31     printf("%d\n", f5(3, 15));
32     printf("%d\n", f6(1, 1));
33     printf("%d\n", f6(3, 2));
34     return 0;
35 }
```

請說明程式執行後，程式碼編號26~33的輸出，以及其運算邏輯。(25分)

二、關於以下C++程式碼：

```
01 #include <iostream>
02 #include <string>
03 #include <exception>
04 #include <stdexcept>
05 #include <assert.h>
06 using namespace std;
07 class NegativeException: public exception {
08     const char * what () const throw () { return "negative"; }
09 };
10 class DivideByZeroException: public logic_error{
11 public:
12     DivideByZeroException() : logic_error( "divide by zero" ) {}
13 };
14 int getResult(int x, int y) {
15     if (x<0 || y<0) throw NegativeException();
16     else if (y==0) throw DivideByZeroException();
17     return (x/y);
18 }
19 void f(int x, int y) {
20     try { cout << "Result:"<< getResult(x, y)<<endl; }
21     catch (std::exception &e) { cout << "1: " << e.what() << "\n"; }
22 }
23 void testResult() {
24     f(2, -1);
25     f(2, 0);
26     f(2, 3);
27     f(6, 3);
28 }
29 void assertResult() {
30     assert(getResult(8, 4)==1);
31 }
32 int main() {
33     testResult();
34     assertResult();
35     return 0;
36 }
```

(一)請說明程式執行後的輸出。(15分)

(二)請說明程式中assert與exception的使用時機與目的。(10分)

## 三、關於以下Java程式：

```

01 import java.io.*;
02 import java.util.ArrayList;
03 abstract class Fruit {
04     public Fruit(int sweetness) { this.sweetness = sweetness; }
05     public abstract String eat();
06     protected String taste() {
07         if (sweetness>0 && sweetness<5) return "no";
08         else if (sweetness<=10) return "little";
09         else if (sweetness<=15) return "some";
10         else if (sweetness<=20) return "more";
11         else return "super";
12     }
13     private int sweetness;
14 }
15 class Apple extends Fruit {
16     public Apple(String c, int s) { super(s); this.color = c; }
17     public String eat() { return color + ":" + taste() + " sweetness "; }
18     private String color;
19 }
20 class Watermelon extends Fruit {
21     public Watermelon(String v, int s) { super(s); this.volume = v; }
22     public String eat() { return volume + ":" + taste() + " sweetness "; }
23     private String volume;
24 }
25 public class Test {
26     public static void test01() {
27         ArrayList<Fruit> fs = new ArrayList<Fruit>();
28         fs.add(new Apple("red", 18));
29         fs.add(new Watermelon("big", 20));
30         fs.add(new Apple("green", 10));
31         fs.forEach((n) -> System.out.println(n.eat()));
32     }
33     public static void main(String[] args) throws InterruptedException {
34         test01();
35     }
36 }

```

(一)請說明程式執行後其輸出與其運作程式碼行數順序。(12分)

(二)請依下面表格，說明Fruit的設計功用，包含Fruit類別類型與功用以及方法 (method)。(13分)

	Fruit類別類型與功用	Fruit方法功用	eat功用	taste功用
說明				

四、以下C++程式有部分違反安全程式設計原則，可能具有許多潛在風險。

```
01 #include <iostream>
02 #include <string>
03 #define SIZE 10
04 using namespace std;
05 class Food {
06 public:
07     Food() = default;
08     Food(int c) { cal = c; }
09     int getCal() { return cal; }
10 private:
11     int cal;
12 };
13 void f1() {
14     Food *f[SIZE];
15     cout<<f[0]->getCal()<<endl;
16 }
17 void f2(int n) {
18     string *f = NULL;
19     for(int i = 0; i < n; i++)
20         f = new string("ok");
21     cout<<*f<<endl;;
22 }
23 void f3(int n) {
24     double x = 3, y1 = 5, y2 = 2;
25     for (int i=0; i<n; i++) {
26         x = x/10.0;
27         y1 = y1/10.0; y2 = y2/10.0;
28     }
29     if(x == (y1-y2)) cout<<"X == Y"<<endl;
30 }
31 void f4(char *s1, char *s2) {
32     int len =0;
33     char *s =s1 ;
34     while (*s2!='\0') {
35         *s1=*s2;
36         s1++; s2++;
37     }
38     cout<<s<<endl;
39 }
40 void f5(int n) {
41     int result = 0;
42     int *d = new int[n];
43     d[0] = d[1] = 1;
44     for(int i = 0; i < n-2; i++) d[i+2] = d[i+1]+d[i];
45     for(int i = 0; i < n; i++) result = result + d[i];
46     cout<<result<<endl;
47 }
48 int main() {
49     char s1 []="goodness", s2 []="food";
50     f1();
51     f2(2);
52     f3(1);
53     f4(s1, s2);
54     f5(6);
55     return 0;
56 }
```

請說明此程式，執行函式f1()~f5()的輸出，以及函式f1()~f5()可能具有的潛在風險。(25分)