

# 《程式設計》

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)

## 試題評析

本題測試C語言的基本語法，主要針對基本型態與陣列兩種用法作為測試主題，再利用函數分別提供參數呼叫產生結果，只要小心作答就能得分。

## 考點命中

《高點·高上程式設計講義》第一回，許振明編撰，頁9-14，45-48。

答：

指令	執行結果	說明
printf("%u\n", f1(6, 2));	0	$6 \&\& !2 = 6 \&\& 0 = 0$
printf("%u\n", f2(6, 2));	24	$6 \ll 2 = 24$ ，6的二進位左移2個位元得到24
printf("%u\n", f3(7, 2));	5	$7 \&\sim 2 = 5$ ，相當於遮掉7右邊數過來的第2位元，相當於7減2得到5
printf("%d\n", f4(3, 12));	27	$3*12+3-12 = 27$
printf("%d\n", f5(15, 3));	155	$f\ 5(15,3) = data[3] = f4(12,13) = 12*13+12-13 = 155$
printf("%d\n", f5(3, 15));	0	$f\ 5(3,15) = data[15]$ ，已超出陣列範圍，如果沒發生segmentation fault，就會印出在data[15]計算出來的位址上的亂數值
printf("%d\n", f6(1, 1));	4	$f\ 6(1,1) = data[1][1] = 4$
printf("%d\n", f6(3, 2));	3	$f\ 6(3,2) = data[3][2] = 3$

## 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)

試題評析	本題測試C++語言的Exception處理機制，先自行定義Exception子物件，再利用Exception子物件處理判斷是否有負值與分母為0的情況，最後再使用assert測試執行結果是否符合預期的答案，物件導向Exception是基本用法，仔細作答可得高分。
考點命中	《高點·高上程式設計講義》第二回，許振明編撰，頁50-53。

**答：**

(一)

1: negative

1: divide by zero

Result:0

Result:2

Assertion failed!

(二)

assert: 測試指定的輸入下是否得到預計的結果，如

assert(getResult(8, 4)==1);

測試getResult(8, 4)的結果是否為1，getResult(8, 4)=2，結果測試失敗。

exception: 指定的程式區塊(try{})中，程式的執行是否有預估的狀況發生(catch{})，如果有，則拋出指定的Exception進行後續的處理。如

if (x<0 || y<0) throw NegativeException();

// x或y有小於0，拋出NegativeException

else if (y==0) throw DivideByZeroException(); // 分母為0，拋出DivideByZeroException

三、關於以下Java程式

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	<pre> import java.io.*; import java.util.ArrayList; abstract class Fruit {     public Fruit(int sweetness) {this.sweetness = sweetness; }     public abstract String eat();     protected String taste() {         if (sweetness&gt;0 &amp;&amp; sweetness&lt;5) return "no";         else if (sweetness&lt;=10) return "little";         else if (sweetness&lt;=15) return "some";         else if (sweetness&lt;=20) return "more";         else return "super";     }     private int sweetness; } class Apple extends Fruit {     public Apple(String c, int s) { super(s); this.color = c; }     public String eat() { return color + ":"+taste()+" sweetness "; }     private String color; } class Watermelon extends Fruit {     public Watermelon(String v, int s) { super(s); this.volume = v; }     public String eat() { return volume + ":"+taste()+" sweetness "; }     private String volume; } public class Test { </pre>
--	---

```

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)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				

<b>試題評析</b>	本試題測試物件導向抽象類別的用法，由具體子類別(Apple, Watermelon)實作繼承於父類別(Fruit)的抽象函數(eat())，再使用ArrayList儲存子物件後，使用迴圈呼叫所有的子物件執行eat()函數並印出結果，用功的同學容易得高分。
<b>考點命中</b>	《高點·高上程式設計講義》第一回，許振明編撰，頁119-123。

**答：**

(一)

red:more sweetness  
big:more sweetness  
green:little sweetness

列號	說明
33	進入主程式main()
34	呼叫test01()函數
26	進入test01()函數
27	建立ArrayList<Fruit> fs
28	新增物件Apple("red",18)並加入ArrayList<Fruit> fs
16	呼叫Apple物件建構子並進行初始化
04	呼叫Fruit物件建構子並進行初始化
29	新增物件Watermelon("big",20)並加入ArrayList<Fruit> fs
21	呼叫Watermelon物件建構子並進行初始化
04	呼叫Fruit物件建構子並進行初始化
30	新增物件Apple("green",10)並加入ArrayList<Fruit> fs
16	呼叫Apple物件建構子並進行初始化
04	呼叫Fruit物件建構子並進行初始化
31	將ArrayList<Fruit> fs內的物件逐一取出並執行迴圈
17	執行Apple物件內的eat()函數
17	執行Apple物件內的eat()函數
35	回到主程式後結束程式

(二)

	Fruit類別類型與功用	Fruit方法功用	eat功用	taste功用
說明	1. 抽象類別 Fruit 2. 建構子 Fruit()初始化 sweetness 成員 3. 提供私用成員 sweetness 紀錄甜度 4. 抽象函數 eat() 5. 具體函數 taste()	1. 抽象函數 eat()顯示水果吃起來的狀況，包含甜度、顏色與容量等資訊，由子類別自行定義內容。 2. 具體函數 taste()根據甜度係數顯示甜度	類別Fruit提供抽象函數，子類別 Apple實作eat()顯示顏色與甜度。 子類別Watermelon實作eat()顯示容量與甜度。	taste()根據甜度係數顯示甜度 " no" 、" little" 、" some" 、" more" 或" super" 。

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)

<b>試題評析</b>	本試題主要測試指標(pointer)與字串的操作方法與規則，指標使用時必須注意初始化的問題，並且要小心使用指標前必須確認已經控制物件或變數，否則容易出現錯誤或者Exception。另外，函數也必須注意檢查接收到的參數是否符合要求，小心作答可得高分。
<b>考點命中</b>	《高點·高上程式設計講義》第一回，許振明編撰，頁74-78。

**答：**

(一)

686588744

ok

X == Y

foodness

20

(二)

```
void f1() {
```

```
    Food *f[SIZE];           // 指標陣列f未給預設值
```

```
    cout<<f[0]->getCal()<<endl; // f[0]未設定物件，呼叫getCal()函數可能出現Exception
```

```
}
```

```
void f2(int n) {
```

```
    string *f = NULL;
```

```
    // 執行迴圈不斷產生字串物件"ok"給字串指標f，可能造成大量垃圾物件
```

```
    for(int i = 0; i < n; i++)
```

```
        f = new string("ok");
```

```
    cout<<*f<<endl;
```

```
}
```

```
void f3(int n) {
```

```
    double x = 3, y1 = 5, y2 = 2;
```

```
    // 利用迴圈連續將x,y1,y2除以10.0後，如果n值很大，可能造成x,y1,y2均為0，最後if條件均為真
```

```
    for (int i=0; i<n; i++) {
```

```
        x = x/10.0;
```

```
        y1 = y1/10.0; y2 = y2/10.0;
```

```
    }
```

```
    if(x == (y1-y2)) cout<<"X == Y"<<endl;
```

【版權所有，重製必究！】

```
}  
void f4(char *s1, char *s2) {  
    int len = 0;  
    char *s = s1 ;  
    // 執行迴圈後，未將'\0'設給s1，如果s2長度大於s1，可能造成s1沒有字串結尾符號'\0'  
    while (*s2!='\0') {  
        *s1=*s2;  
        s1++; s2++;  
    }  
    cout<<s<<<endl;  
}  
void f5(int n) {  
    int result = 0;  
    int *d = new int[n];    // 未檢查n值是否大於0，可能造成配置記憶體錯誤  
    d[0] = d[1] = 1;  
    for(int i = 0; i < n-2; i++) d[i+2] = d[i+1]+d[i];  
    for(int i = 0; i < n; i++) result = result + d[i];  
    cout<<result<<endl;  
}
```

【版權所有，重製必究！】