# 112年公務人員高等考試三級考試試題

類　　科：資訊處理
科　　目：程式設計
考試時間：２小時　　　　　　　　　　　　　　　　　　座號：＿＿＿＿＿＿

※注意：㈠禁止使用電子計算器。
㈡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。
㈢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：

```
#include <iostream>
#include <string>
#include <exception>
#include <stdexcept>
#include <assert.h>
using namespace std;
class NegativeException: public exception {
    const char * what () const throw () { return "negative"; }
};
class DivideByZeroException: public logic_error{
public:
    DivideByZeroException() : logic_error( "divide by zero" ) {}
};
int getResult(int x, int y) {
    if (x<0 || y<0) throw NegativeException();
    else if (y==0) throw DivideByZeroException();
    return (x/y);
}
void f(int x, int y) {
    try { cout << "Result:"<< getResult(x, y)<<endl; }
    catch (std::exception &e) { cout << "1: " << e.what() << "\n"; }
}
void testResult() {
    f(2, -1);
    f(2, 0);
    f(2, 3);
    f(6, 3);
}
void assertResult() {
    assert(getResult(8, 4)==1);
}
int main() {
    testResult();
    assertResult();
    return 0;
}
```

㈠請說明程式執行後的輸出。（15分）

㈡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  }
```

㈠請說明程式執行後其輸出與其運作程式碼行數順序。（12分）

㈡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。

```
#include <iostream>
#include <string>
#define SIZE 10
using namespace std;
class Food {
public:
    Food() = default;
    Food(int c) { cal = c; }
    int getCal() { return cal; }
private:
    int cal;
};
void f1() {
    Food *f[SIZE];
    cout<<f[0]->getCal()<<endl;
}
void f2(int n) {
    string *f = NULL;
    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;
    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;
}
void f4(char *s1, char *s2) {
    int len =0;
    char *s =s1 ;
    while (*s2!='\0') {
        *s1=*s2;
        s1++; s2++;
    }
    cout<<s<<endl;
}
void f5(int n) {
    int result = 0;
    int *d = new int[n];
    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;
}
int main() {
    char s1[]="goodness", s2[]="food";
    f1();
    f2(2);
    f3(1);
    f4(s1, s2);
    f5(6);
    return 0;
}
```

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）