

0. 名詞解釋與關鍵字功用。(6%) +5

(0) 主動(顯性)型態轉換:

又稱強制型態轉換,
 發生在希望輸出型態不同時

ex: float a;
 cout << (int) a;

(1) 語意錯誤:

程式語法正確, 但是輸出
 結果不符合預期

(2) 函數多載:

多載 (overloading) 是指在程式中
 可以定義相同函數名稱, 但具有不
 同的引數個數或型態, 以實現不同
 的功能

2. int 整數最大值加 1 為何會變最小值? (2%)

因為 int 佔 4 個位元組, 若是
 最大值加 1, 會發生溢位, 而
 其負數是用 2 的補數法表
 示最後就變成最小值

1. 說明以下名詞或關鍵字之差別。(6%) +6

(0) "=" 與 "==" :

= : 設定變數的值

== : 判定是否真的等於

(1) 無限迴圈與空迴圈:

無限迴圈: 結束條件不成立或沒有設
 置結束條件使迴圈一直執行

空迴圈: 執行迴圈, 但是不會輸
 出結果, 或是最後才輸出結果

(2) stdlib.h 與 cstdlib:

都是標頭檔

stdlib.h : 用 C 語言時所用之標頭檔

cstdlib : 用 C++ 語言時所用之標頭檔

3. 多個函數彼此呼叫時, 撰寫順序該如何安排? (2%) +2

int area(int i)

{
 }

int len(int j)

{
 area(i);
 }

len 函數呼叫 area
 函數, 所以 area 函數
 要寫在 len 函數前
 面

4. 請回答利用 printf 函數以下列格式列印之
 結果? 請自行繪製格子並填入其中。(4%)

資料內容 格式 執行結果 +4

(0) 369 %06d ±00369 ✓

(1) 8.26 %-6.1f 8.2 --- ✓

5. 下列陣列初始化方式語法是否正確? (4%)

(0) (X) int arr[3] = {1, 2, 3, 4}; +4

(1) (O) int arr[3] = {1, 2};

(2) (O) int arr[] = {1.5, 2, 3.14};

(3) (O) int arr[] = {0};

6. 試判別下列的敘述之執行結果。(6%) +6

(0) 6 + (4 == 9) 答: 6

(1) 7 > 0 || 6 < 6 || 12 < 13 答: true

(2) 3 <= 6 ? 3 / 6 : 6 / 3 答: 0

(3) !5 + !0 答: 1

(4) ~12 + 12 答: -1

(5) ('d' - 'a') % 2 == 1 ? 1 : 0 答: 1

7. int a=10, b=5, 求下列之 a、b 結果。(6%)

(0) b %= a - b b = b % (a - b) a = 10 b = 0 +4.5

(1) a += ++b / a a = a + (++b) / a a = 10 b = 6 => 5 + 1 = 6

(2) b %= !(-a) % (-b) a = 10 b = 5

(3) a *= (b++) + b a = 100 b = 6

(4) b /= -a + ++b a = 9 b = 2

(5) a %= ++a > b++ a = 0 b = 6

a = a % (++a > b++)
 10 % 1 = 0

8. 請問以下程式輸出為何? (12%)

```
for(int i=9; i>1; i/=2)
{
    for(int j=1; j>0; j--=2)
    {
        cout << j;
        printf("\n");
    }
}
```

(3%) Output
9 7 5 3 1
4 2
2

```
int f(int a, int y) {
    cout << a++ << --y;
    return a;
}

int main(void) {
    int x=6, y=3;
    f(x, y) /= y;
    cout << x << endl;
}
```

(3%) Output
6 2 2

```
int i;
for(i=1; i<=3; i++)
{
    cout << "0";
    cout << "1";
}
cout << endl;
for(i=1; i<=3; i++)
{
    cout << "0";
    cout << "1";
}
cout << endl;
for(i=1; i<=3; i++)
{
    cout << "0";
    cout << "1";
}
```

(3%) Output
010101
0001
01

```
int i=0;
for(i=-1; i<10; i++)
{
    printf("%d\n", i);
    do
    {
        i++;
    } while(i<20);
    printf("%d\n", i);
    while(i>0)
    {
        i++;
    }
    printf("%d\n", i);
}
```

(3%) Output
11
10 20
11

1 0 -1 -2 ... 最小值
int 最大值+1 = int 最小值

9. 將下列空格填入正確的程式碼。(12%)

```
/*輸入數字n，印出1-100間，n之所有倍數*/

#include <iostream>
#include <cstdlib>
using namespace std;
int main(void)
{
    int n, i=1; //宣告整數變數
    <2> >> n; //從鍵盤讀入一個整數存到n中
    while(<3>) //迴圈從1跑到100
    {
        if(<4>) //檢查i是否為n的倍數
            cout << i; //印出該倍數
        <5> i++;
    }
    system("pause");
    return 0;
}
```

0. std	1. main	2. cin
3. i<=100	4. i%n==0	5. i++;

/*計算1+2+4+8+16之結果並且印出*/

```
#include <stdio.h>
#include <stdlib.h>
int main(void) //無須作答
{
    int i, sum; //宣告整數變數
    for(i=1; i<=16; i++) //迴圈跑1 2 4 8 16
        sum += i; //加總
    printf("%d", sum); //印出總和
    system("pause");
    return 0;
}
```

6. sum=0	7. i=1	8. i<=16
9. i*=2	10. i++	11. sum

10. 上機實作題 (40%) //無須作答，考試結束後由老師填入

題號	Q1	Q2	Q3	Q4
得分	10	10	2	0

Q1. 長方體之最大面積計算 (10%)

說明:長方體由長、寬、高組成，共有六個表面，求六面中的最大表面積。例如長=3、寬=2、高=4之長方體，其表面積分別為 6、12、8，其中最大表面積為 12。

輸入:先輸入 1 個正整數 n ($n \leq 20$)，接著輸入 n 組長方體資料 l 、 w 、 h ($1 \sim 100$) 代表長方體之長寬高，數字之間以空白隔開。

輸出: n 行長方體最大表面積計算之結果 (記得換行)。

範例資料:

輸入	輸出
3	12
3 2 4	6
1 2 3	25
5 5 5	

Q2. 零錢兌換 (10%)

說明:我國硬幣分為 1 元、5 元、10 元與 50 元硬幣，給定一個金額數字 m ，求金額 m 最少須幾枚硬幣組成，例如 12 元可由 1 個 10 元與 2 個 1 元所組成，最少須 3 枚硬幣。

輸入:先輸入 1 個正整數 n ($n \leq 20$)，接著輸入 n 組金額正整數 m ($0 \leq m \leq 1000$)。

輸出:金額 m 最少須幾枚硬幣組成 (記得換行)。

範例資料:

輸入	輸出
3	3
12	7
87	8
168	

Q3. 分數相加 (10%)

說明:輸入的四個整數分別代表兩個分數 a/b 與 c/d ，請將兩個分數通分後，進行相加，再進行約分，並且將結果輸出。結果若為假分數無須化為整數或帶分數。舉例來說，若結果為 $2/1$ ，無須化為 2，以第一組輸入資料為例， $1/2 + 3/4$ 為 $5/4$ 。

輸入:先輸入 1 個正整數 n ($n \leq 20$)，接著輸入 n 組四個正整數 a 、 b 、 c 、 d ($1 \sim 100$)，數字之間以空白隔開。

輸出:將 $a/b + c/d$ 之結果約分之後，以假分數形式輸出 (記得換行)。

範例資料:

輸入	輸出
2	5/4
1 2 3 4	1/2
3 8 1 8	

$$\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$$

Q4. 鏡數判斷 (10%)

說明:猶如數字照鏡子一般，當一個數字從左至右看與從右至左看，數值都一樣時，被稱為鏡數，例如 121、5665、777 等都屬於鏡數。

輸入:先輸入 1 個正整數 n ($n \leq 20$)，接著輸入 n 組正整數 m 代表欲判斷之數字 ($m \leq 2^{31}-1$)。

輸出:若數字 m 為鏡數則輸出 "Yes"，否則輸出 "No" (記得換行)。

測試資料:

輸入	輸出
3	Yes
121	No
123	Yes
5665	

$$\begin{array}{r} 1000 \\ - 121 \\ \hline 879 \end{array} \quad \begin{array}{r} 10000 \\ - 5665 \\ \hline 4335 \end{array} \quad \begin{array}{r} 10000000 \\ 1134321 \\ \hline 8865679 \end{array} \quad 123$$