

1. 給定右側函式  $f()$ ，當執行  $f(10)$  時，最終回傳結果為何？

- (A) 1
- (B) 3840
- (C) -3840
- (D) 執行時導致無窮迴圈，不會停止執行

```
int f (int i) {  
    if (i>0)  
        if ((i/2)%2)==0  
            return f(i-2)*i;  
        else  
            return f(i-2)*(-i);  
    else  
        return 1;  
}
```

2. 給定右側程式，若已知輸出的結果為 [1][2][3][5][4][6]，程式中的 (?) 應為下列何者？

- (A)  $j < i$
- (B)  $j > i$
- (C)  $j \leq i$
- (D)  $j \geq i$

```
int main() {  
    int i, j;  
    for (i=0; i<5; i=i+1) {  
        for (j=0; (?); j=j+2)  
            printf("[%d]", i+j);  
    }  
}
```

3. 給定右側函式  $f()$ ，已知  $f(14)$ 、 $f(10)$ 、 $f(6)$  分別回傳 25、18、10，函式中的 (?) 應為下列何者？

- (A)  $(n+1)/2$
- (B)  $n/2$
- (C)  $(n-1)/2$
- (D)  $(n/2)+1$

```
int f(int n) {  
    if (n < 2) {  
        return n;  
    }  
    else {  
        return (n + f((?)));  
    }  
}
```

4. 給定右側程式片段，當程式執行完後，輸出結果為何？

- (A) 9
- (B) 18**
- (C) 27
- (D) 30

```
int Q[200];
int i, val=0;
int count=0;
int head=0, tail=0;
for (i=1; i<=30; i=i+1) {
    Q[tail] = i;
    tail = tail + 1;
}
while (tail > head + 1) {
    val = Q[head];
    head = head + 1;
    count = count + 1;
    if (count == 3) {
        count = 0;
        Q[tail] = val;
        tail = tail + 1;
    }
}
printf ("%d", Q[head]);
```

5. 右側程式正確的輸出應該如下：

```
      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * *
```

在不修改右側程式之第4行及第7  
行程式碼的前提下，最少需修改  
幾行程式碼以得到正確輸出？

```
1 int k = 4;
2 int m = 1;
3 for (int i=1; i<=5; i=i+1) {
4     for (int j=1; j<=k; j=j+1) {
5         printf (" ");
6     }
7     for (int j=1; j<=m; j=j+1) {
8         printf ("*");
9     }
10    printf ("\n");
11    k = k - 1;
12    m = m + 1;
13 }
```

- (A) 1
- (B) 2
- (C) 3
- (D) 4

6. 給定一整數列表  $a[0]$ 、 $a[1]$ 、...、 $a[99]$ 且  $a[k]=3k+1$ ，以  $value=100$  呼叫以下兩函式，假設函式 **f1** 及 **f2** 之 **while** 迴圈主體分別執行  $n1$  與  $n2$  次（也就是說計算 **if** 敘述執行次數，不包含 **elif** 敘述），請問  $n1$  與  $n2$  之值為何？註： $(low + high)/2$  只取整數部分。

```
int f1(int a[], int value) {
    int r_value = -1;
    int i = 0;
    while (i < 100) {
        if (a[i] == value) {
            r_value = i;
            break;
        }
        i = i + 1;
    }
    return r_value;
}
```

```
int f2(int a[], int value) {
    int r_value = -1;
    int low = 0, high = 99;
    int mid;
    while (low <= high) {
        mid = (low + high)/2;
        if (a[mid] == value) {
            r_value = mid;
            break;
        }
        else if (a[mid] < value) {
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return r_value;
}
```

- (A)  $n1=33, n2=4$   
(B)  $n1=33, n2=5$   
(C)  $n1=34, n2=4$   
(D)  $n1=34, n2=5$

7. 一個費式數列定義第一個數為 0，第二個數為 1，之後的每個數都等於前兩個數相加，如下所示：

0、1、1、2、3、5、8、13、21、34、55、89...。右列的程式用以計算並印出第 N 個 (N≥2)費式數列的數值，請問 (a) 與 (b) 兩個空格的敘述 (statement) 應該為何？

```
int a=0;
int b=1;
int i, temp, N;
...
for (i=2; i<=N; i=i+1) {
    temp = b;
    (a) ;
    a = temp;
    printf ("%d\n", (b) );
}
```

- (A) (a)  $f[i] = f[i-1]+f[i-2]$       (b)  $f[N]$
- (B) (a)  $a = a+b$       (b)  $a$
- (C) (a)  $b = a+b$       (b)  $a$
- (D) (a)  $f[i] = f[i-1]+f[i-2]$       (b)  $f[i]$

8. 給定右側函式 **f1( )** 及 **f2( )**。 **f1(1)** 運算過程中，以下敘述何者為錯？

- (A) 印出的數字最大的是 4
- (B) **f1** 一共被呼叫二次
- (C) **f2** 一共被呼叫三次
- (D) 數字 2 被印出兩次

```
void f1 (int m) {
    if (m > 3) {
        printf ("%d\n", m);
        return;
    }
    else {
        printf ("%d\n", m);
        f2(m+2);
        printf ("%d\n", m);
    }
}

void f2 (int n) {
    if (n > 3) {
        printf ("%d\n", n);
        return;
    }
    else {
        printf ("%d\n", n);
        f1(n-1);
        printf ("%d\n", n);
    }
}
```

9. 右側程式片段擬以輾轉除法求  $i$  與  $j$  的最大公因數。請問 **while** 迴圈內容何者正確？

(A)  $k = i \% j$

$i = j$

$j = k$

(B)  $i = j$

$j = k$

$k = i \% j$

(C)  $i = j$

$j = i \% k$

$k = i$

(D)  $k = i$

$i = j$

$j = i \% k$

```

i = 76;
j = 48;

while ((i % j) != 0) {
    _____
    _____
    _____
}

printf ("%d\n", j);
    
```

10. 右側程式執行過後所輸出數值為何？

- (A) 11
- (B) 13
- (C) 15
- (D) 16

```
void main () {
    int count = 10;
    if (count > 0) {
        count = 11;
    }
    if (count > 10) {
        count = 12;
        if (count % 3 == 4) {
            count = 1;
        }
        else {
            count = 0;
        }
    }
    else if (count > 11) {
        count = 13;
    }
    else {
        count = 14;
    }
    if (count) {
        count = 15;
    }
    else {
        count = 16;
    }

    printf ("%d\n", count);
}
```

11. 右側為一個計算 n 階乘的函式，請問  
該如何修改才會得到正確的結果？

```
1. int fun (int n) {  
2.     int fac = 1;  
3.     if (n >= 0) {  
4.         fac = n * fun(n - 1);  
5.     }  
6.     return fac;  
7. }
```

- (A) 第2行，改為 **fac = n**  
(B) 第3行，改為 **if (n > 0) {**  
(C) 第4行，改為 **fac = n \* fun(n+1)**  
(D) 第4行，改為 **fac = fac \* fun(n-1)**
12. 右側 **f( )** 函式 **(a)** , **(b)** , **(c)** 處需  
分別填入哪些數字，方能使得 **f(4)** 輸  
出 2468 的結果？

```
int f(int n) {  
  
    int p = 0;  
  
    int i = n;  
  
    while (i >= _____ (a) ) {  
  
        p = 10 - _____ (b) * i;  
  
        printf ("%d", p);  
  
        i = i - _____ (c) ;  
  
    }  
  
}
```

- (A) **1, 2, 1**  
(B) **0, 1, 2**  
(C) **0, 2, 1**  
(D) **1, 1, 1**

13. 右側 **g(4)** 函式呼叫執行後，回傳值為何？

- (A) 6
- (B) 11
- (C) 13
- (D) 14

```
int f (int n) {  
    if (n > 3) {  
        return 1;  
    }  
    else if (n == 2) {  
        return (3 + f(n+1));  
    }  
    else {  
        return (1 + f(n+1));  
    }  
}  
  
int g(int n) {  
    int j = 0;  
    for (int i=1; i<=n-1; i=i+1) {  
        j = j + f(i);  
    }  
    return j;  
}
```

14. 右側 **Mystery( )** 函式 **else** 部分  
運算式應為何，才能使得  
**Mystery(9)** 的回傳值為 **34**。

- (A)  $x + \text{Mystery}(x-1)$
- (B)  $x * \text{Mystery}(x-1)$
- (C)  $\text{Mystery}(x-2) + \text{Mystery}(x+2)$
- (D)  $\text{Mystery}(x-2) + \text{Mystery}(x-1)$

```
int Mystery (int x) {  
    if (x <= 1) {  
        return x;  
    }  
    else {  
        return _____ ;  
    }  
}
```

15. 右側程式碼執行後，  
輸出結果為何？

- (A) 2468975319
- (B) 1357924689
- (C) 1234567899
- (D) 2468513799

```
int a[9] = {1, 3, 5, 7, 9, 8, 6, 4, 2};
int n=9, tmp;

for (int i=0; i<n; i=i+1) {
    tmp = a[i];
    a[i] = a[n-i-1];
    a[n-i-1] = tmp;
}
for (int i=0; i<=n/2; i=i+1)
    printf ("%d %d ", a[i], a[n-i-1]);
```

16. 右側函式以 **F(7)** 呼叫後回傳值為  
12，則 **<condition>** 應為何？

- (A)  $a < 3$
- (B)  $a < 2$
- (C)  $a < 1$
- (D)  $a < 0$

```
int F(int a) {
    if ( <condition> )
        return 1;
    else
        return F(a-2) + F(a-3);
}
```

17. 右側是依據分數  $s$  評定等第的程式碼片段，

正確的等第公式應為：

90~100 判為 A 等

80~89 判為 B 等

70~79 判為 C 等

60~69 判為 D 等

0~59 判為 F 等

這段程式碼在處理 0~100 的分數時，有幾個分數的等第是錯的？

- (A) 20
- (B) 11**
- (C) 2
- (D) 10

```
if (s>=90) {  
    printf ("A \n");  
}  
else if (s>=80) {  
    printf ("B \n");  
}  
else if (s>60) {  
    printf ("D \n");  
}  
else if (s>70) {  
    printf ("C \n");  
}  
else {  
    printf ("F\n");  
}
```

18. 右側程式  
片段執行  
後，**count**  
的值為何？

(A) 36

(B) 20

(C) 12

(D) 3

```
int maze[5][5]= {{1, 1, 1, 1, 1},
                 {1, 0, 1, 0, 1},
                 {1, 1, 0, 0, 1},
                 {1, 0, 0, 1, 1},
                 {1, 1, 1, 1, 1}};

int count=0;
for (int i=1; i<=3; i=i+1) {
    for (int j=1; j<=3; j=j+1) {
        int dir[4][2] = {{-1,0}, {0,1}, {1,0}, {0,-1}};
        for (int d=0; d<4; d=d+1) {
            if (maze[i+dir[d][0]][j+dir[d][1]]==1) {
                count = count + 1;
            }
        }
    }
}
```

19. 右側程式片段中執行後若要印出下列圖案，(a) 的條件判斷式該如何設定？

```
* * * * *
* * * *
* *
```

```
for (int i=0; i<=3; i=i+1) {
    for (int j=0; j<i; j=j+1)
        printf(" ");
    for (int k=6-2*i; (a) ; k=k-1)
        printf("*");
    printf("\n");
}
```

- (A)  $k > 2$
- (B)  $k > 1$
- (C)  $k > 0$
- (D)  $k > -1$

20. 下列程式碼是自動計算找零程式的一部分，程式碼中三個主要變數分別為 **Total**(購買總額)，**Paid** (實際支付金額)，**Change**(找零金額)。但是此程式片段有冗餘的程式碼，也就是移除該段程式碼之後不會影響程式的功能。請找出冗餘程式碼的區塊。

(A) 冗餘程式碼  
在 A 區

(B) 冗餘程式碼  
在 B 區

(C) 冗餘程式碼  
在 C 區

(D) 冗餘程式碼  
在 D 區

```
int Total, Paid, Change;
...
Change = Paid - Total;
printf ("500 : %d pieces\n", (Change-Change%500)/500);
Change = Change % 500;

printf ("100 : %d coins\n", (Change-Change%100)/100);
Change = Change % 100;

// A 區
printf ("50 : %d coins\n", (Change-Change%50)/50);
Change = Change % 50;

// B 區
printf ("10 : %d coins\n", (Change-Change%10)/10);
Change = Change % 10;

// C 區
printf ("5 : %d coins\n", (Change-Change%5)/5);
Change = Change % 5;

// D 區
printf ("1 : %d coins\n", (Change-Change%1)/1);
Change = Change % 1;
```

21. 右側  $G()$  為遞迴函式， $G(3, 7)$  執行後回傳值為何？

- (A) 128
- (B) 2187
- (C) 6561
- (D) 1024

```
int G (int a, int x) {
    if (x == 0)
        return 1;
    else
        return (a * G(a, x - 1));
}
```

22. 給定一個  $1 \times 8$  的列表  $A$ ， $A = [0, 2, 4, 6, 8, 10, 12, 14]$ 。右側函式  $Search(x)$  真正目的是找到  $A$  之中大於  $x$  的最小值。然而，這個函式有誤。請問下列哪個函式呼叫可測出函式有誤？

- (A)  $Search(-1)$
- (B)  $Search(0)$
- (C)  $Search(10)$
- (D)  $Search(16)$

```
int A[8]={0, 2, 4, 6, 8, 10, 12, 14};

int Search(int x){
    int high= 7;
    int low = 0;
    while(high > low){
        int mid = (high + low)/2;
        if(A[mid] <= x){
            low = mid + 1;
        }
        else {
            high = mid;
        }
    }
    return A[high];
}
```

23. 右側函式兩個回傳式分別該如何撰寫，才能正確計算並回傳兩參數  $a$ ,  $b$  之最大公因數 (Greatest Common Divisor) ?

- (A)  $a$ ,  $\text{GCD}(b, r)$
- (B)  $b$ ,  $\text{GCD}(b, r)$
- (C)  $a$ ,  $\text{GCD}(a, r)$
- (D)  $b$ ,  $\text{GCD}(a, r)$

```
int GCD (int a, int b){  
  
    int r;  
  
    r = a % b  
    if r == 0:  
        return _____  
    return _____  
  
}
```

24. 若  $A$  是一個可儲存  $n$  筆整數的列表，且資料儲存於  $A[0] \sim A[n-1]$ 。經過右側程式碼運算後，以下何者敘述不一定正確？

- (A)  $p$  是  $A$  列表資料中的最大值
- (B)  $q$  是  $A$  列表資料中的最小值
- (C)  $q < p$
- (D)  $A[0] \leq p$

```
int A[n]={ ... };  
int p = q = A[0];  
for (int i=1; i<n; i=i+1) {  
  
    if (A[i] > p)  
        p = A[i];  
  
    if (A[i] < q)  
        q = A[i];  
  
}
```

25. 右側 **F** ( ) 函式執行時，若輸入依序為整數 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，請問 **X** 列表的元素值依順序為何？

- (A) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- (B) 2, 0, 2, 0, 2, 0, 2, 0, 2, 0
- (C) 9, 0, 1, 2, 3, 4, 5, 6, 7, 8
- (D) 8, 9, 0, 1, 2, 3, 4, 5, 6, 7

```
void F () {  
    int X[10] = {0};  
    for (int i=0; i<10; i=i+1) {  
        scanf("%d",&X[(i+2)%10]);  
    }  
}
```

26. 右側程式片段無法正確列印 20 次的 "Hi!"，請問下列哪一個修正方式仍無法正確列印 20 次的 "Hi!"？

- (A) 將 `range(0, 101, 5)` 改為 `range(0, 20, 1)`
- (B) 將 `range(0, 101, 5)` 改為 `range(5, 101, 5)`
- (C) 將 `range(0, 101, 5)` 改為 `range(0, 100, 5)`
- (D) 將 `range(0, 101, 5)` 改為 `range(5, 100, 5)`

```
for (int i=0; i<=100; i=i+5 5) {  
    printf ("%s s\n", "Hi!");  
}
```

27. 給定右側函式  $F()$ ，執行  $F()$  時哪一行程式碼永遠不會被執行到？

- (A)  $a = a + 5$
- (B)  $a = a + 2$
- (C)  $a = 5$
- (D) 每一行都執行得到

```
Void F(int a) {  
    while(a < 10)  
        a = a + 5;  
    if(a < 12)  
        a = a + 2;  
    if(a <=11)  
        a = 5;  
}
```

28. 給定右側函式  $F()$ ， $F()$  執行完所回傳的  $x$  值為何？

- (A)  $n(n+1) \sqrt{\log_2 n}$
- (B)  $n^2(n+1)/2$
- (C)  $n(n+1) [\log_2 n + 1]/2$
- (D)  $n(n+1)/2$

```
int F (int n) {  
    int x = 0;  
    for(int i=1; i<=n; i=i+1)  
        for(int j=i; j<=n; j=j+1)  
            for(int k=1; k<=n; k=k*2)  
                x= x+ 1;  
    return x;  
}
```

29. 右側程式擬找出列表 **A** 中的最大值 (**M**) 和最小值 (**N**)。不過，這段程式碼有誤，請問 **A** 初始值如何設定就可以測出程式有誤？

- (A) [90, 80, 100]
- (B) [80, 90, 100]
- (C) [100, 90, 80]
- (D) [90, 100, 80]

```
int main () {  
  
    int M = -1, N = 101, s = 3;  
  
    int A[] = _____?_____;  
  
    for (int i=0; i<s; i=i+1) {  
  
        if (A[i]>M) {  
  
            M = A[i];  
  
        }  
  
        else if (A[i]<N) {  
  
            N = A[i];  
  
        }  
  
    }  
  
    printf("M = %d, N= %d\n", M, N);  
  
    return 0;  
  
}
```

30. 經過運算後，右側程式的輸出為何？

- (A) 1275
- (B) 20
- (C) 1000
- (D) 810

```
for (i=1; i<=100; i=i+1) {  
    b[i] = i;  
}  
  
a[0] = 0;  
  
for (i=1; i<=100; i=i+1) {  
    a[i] = b[i] + a[i-1];  
}  
  
printf ("%d\n", a[50]-a[30]);
```