

1. 試寫 C 『句子』讓使用者輸入日期，驗證使用者是否輸入正確，輸出日數。 [20 分]

執行畫面	使用者輸入	指示：
Q: 日期(mm dd)?	<u>13</u> <u>10</u>	請宣告變數: mm, dd, ndays 若輸入錯誤數字，則顯示錯誤訊息。 設 2 月有 28 日 4、6、9、11 月有 30 日 其餘月份 31 天。 必須使用 『 <u>陣列</u> 』 int maxdays[13];
A:	輸入錯誤!	
Q: 日期(mm dd)?	<u>2</u> <u>30</u>	
A:	輸入錯誤!	
Q: 日期(mm dd)?	<u>3</u> <u>10</u>	
A:	1 月 1 日至 3 月 10 日， 31+28+10 = 69 天	

日期轉換 Date conversion: dd-mm-yyyy to words

Q: enter date (dd-mm-yyyy):	25-12-2005
A:	Dec 25, 2005
January, February, March, April, May, June, July, August, September, October, November, December	

2. 試寫 C 程式，計算以下食物數量 (以件數少為原則)。 [20 分]

食物價錢 (單價)：漢堡包(\$9)、蛋糕(\$12)、蘋果(\$5)、西瓜(\$30)

執行畫面如下：

Q: 付多少錢?	100
A:	\$100 可買 西瓜(\$30)3 個、漢堡包(\$9)1 個，找回 \$1
Q: 付多少錢?	50
A:	\$50 可買 西瓜(\$30)1 個、蛋糕(\$12)1 個、蘋果(\$5)1 個，找回 \$3
...	

提示：本程式與羅馬數字程式類同。

宣告以下陣列： 食物名稱(food[])，單價(price[])
並存入適當的起始值 (按大小次序)

宣告以下變數： 付款(amount)，數量(qty)

程式繼續直至使用者輸入 0 為止。

3. 年終花紅 Year end Bonus

year[]	5	10	15	20
rate[]	3%	5%	10%	15%

執行畫面如下：

Q: 工作年資?	3	Q: 工作年資?	10
A: 年終花紅 = \$0		月薪?	10000
...		A: 年終花紅 = \$10000 × 5% = \$500	

4. 紙幣硬幣 Money (1000,500,100,50,20,10,5,2,1)

note[]	1000	500	100	50	...
freq[]					

```
#define MAX 9
```

```
int num[MAX] = { 1000,500,100,50,20,10,5,2,1};
```

執行畫面如下：

Q: 金額 (1-3000): 1234

A: 1000+100+100+20+10+2+2=1234

或

A: 1000+100×2+20+10+2×2=1234

...

或買東西
後找錢

5. 試寫 C 程式，把『阿拉伯』數字(0-9) 轉為『羅馬』數字 (IVXLCDM)。 [20 分]

arabic[]	1000	900	500	400	...
roman[]	M	CM	D	CD	

```
#define MAX 13
```

```
int num[MAX] = { 1000,900,500,400,100,90,50,40,10,9,5,4,1};
```

```
char roman[MAX][3]
```

```
={"M","DM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};
```

執行畫面如下：

Q: 阿拉伯數字 (1-2000): 2007

A: MMVI

Q: 阿拉伯數字 (1-2000): 1998

A: MDMXCVIII

6. 月份 Month：數字轉文字

maxday[]	31	28	31	30	31	...				
month[]	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	

```
#define MAX 13
```

```
int maxday[MAX] = { 0,31,28,31,30,31,30,31,31,30,31,30,31};
```

```
char month[MAX][10] = {"Jan","Feb","March",
```

```
"April","May","June", "July","Aug","Sep", "Oct","Nov","Dec"};
```

執行畫面如下：

Q: month(1-12): 2

A: Feb 28

Q: month(1-12): 12

A: Dec 31

7. No.of days from Jan 1 up to dd-mm-yy

Q: enter date (dd-mm-yyyy): 25-2-2005

A: $31+25 = 56$ days

Q: enter date (dd-mm-yyyy): 12-5-2005

A: $31+28+31+30+25 = 145$ days

Q: enter date (dd-mm-yyyy): 26-12-2005

A: $31+28+31+30+31+30+31+31+30+31+30+26 = 360$ days

7. 相差日數 nDays:

a. 執行畫面如下：

Q: mm dd: 4 10	Q: mm dd: 1 10
A: $31+28+31+10 = 100$ days	A: 10 days

b. 執行畫面如下：

Q: From date (mm1 dd1): 3 10
To date (mm2 dd2): 6 15
A: 1/1-10/3:
d1 = $31+28+10 = 69$ days
1/1-15/6:
d2 = $31+28+31+30+31+15 = 166$ days
d2-d1 = $166-69 = 97$ days

c. 執行畫面如下：

Q: From date (yy1 mm1 dd1): 2005 3 10
To date (yy2 mm2 dd2): 2007 6 15
A: 1/1-10/3:
d1 = $31+28+10 = 69$ days
1/1-15/6:
d2 = $31+28+31+30+31+15 = 166$ days
$365 \times 2 + 166 - 69 = 827$ days

8. 星期 Day of the week

dow[]

Sun	Mon	Tue	Wed	Thu	Fri	Sat
-----	-----	-----	-----	-----	-----	-----

#define MAX 7

char dow[MAX][4] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

執行畫面如下：

Q: day (0-6): 0	Q: day (0-6): 5
A: Sun, Public holiday.	A: Fri, School day.

9. 只用一句 for 句子，重寫以句子：

printf("A[0]=A[1]; \n"); printf("A[1]=A[2]; \n"); printf("A[2]=A[3]; \n"); printf("A[3]=A[4]; \n"); printf("A[4]=A[5]; \n");	輸出： A[0]=A[1]; A[1]=A[2]; ...
--	--

for(i=0;i<5;i++) printf(...);

10. 星座 12 signs of the Zodiac

```
#define MAX 13
```

```
char zodiac[MAX][5] = {"水瓶","雙魚","白羊","金牛",  
"雙子","巨蟹","獅子","處女","天秤","天蠍","人馬","山羊"};
```

zday[]	20	19	21	21	21	22	23	24	23	23	22	22
zodiac[]	水	魚	羊	牛	子	蟹	獅	女	秤	蠍	馬	山

1	水瓶	Aquarius	01/20-02/18	7	獅子	Leo	07/23-08/22
2	雙魚	Pisces	02/19-03/20	8	處女	Virgo	08/23-09/22
3	白羊	Aries	03/21-04/19	9	天秤	Libra	09/23-10/22
4	金牛	Taurus	04/20-05/20	10	天蠍	Scorpio	10/23-11/21
5	雙子	Gemini	05/21-06/21	11	人馬	Sagittarius	11/22-12/21
6	巨蟹	Cancer	06/22-07/22	12	山羊	Capricorn	12/22-01/19

執行畫面如下：

Q: mm dd: 1 19	Q: mm dd: 1 10
A: 山羊座	A: 水瓶座

方法：取第 mm 個星座 (zd=mm)，若 dd<該星座的首天，則取 mm-1 個星座 (zd--)

11. 生肖 Animals

```
char animal[MAX][3]
```

```
= {"猴","雞","狗","豬","鼠","牛","虎","兔","龍","蛇","馬","羊"};
```

animal[]	猴	雞	狗	豬	鼠	牛	虎	兔	龍	蛇	馬	羊
----------	---	---	---	---	---	---	---	---	---	---	---	---

執行畫面如下：

Q: 出生年份: 2004	Q: 出生年份: 2008
A: 猴年 (2004%12=0)	A: 鼠年 (2008%12=4)

11. Lottery ticket:

```
char luckynumber[3][8]={"1234567","1726354","7654321"};
```

Q: 獎券號碼: 7654321	Q: 獎券號碼: 9999999
A: 三獎	A: 努力工作吧

12. 阿拉伯數字轉中國數字 Arabic to Chinese :

```
int num, digits[10];
```

```
char place[5][3] = {"", "十", "百", "千", "萬"};
```

```
char unit[10][3] = {"〇", "一", "二", "三", "四", "五", "六", "七", "八", "九"};
```

unit[]	〇	一	二	三	四	五	六	七	八	九
place[]		十	百	千	萬					
digits[]	4	3	2	1						

a. 執行畫面如下：

Q: 輸入數字 (0 - 99,999): 1234

A: 一千二百三十四

Q: 輸入數字 (0 - 99,999): 21001

A: 二萬一千〇〇一

b. 執行畫面如下：

Q: 輸入數字 (0 - 99,999,999): 1234 5678

A: 一千二百三十四萬 五千六百七十八

13. 購物結帳 pay & change :

food[]	西瓜	蛋糕	麵包	蘋果
price[]	30	12	9	5
qty[]	1	2	3	0

執行畫面如下：

請輸入食物數量:	應付款: $\$9 \times 1 + \$12 \times 2 + \$5 \times 3 = 48$
麵包(\$9) 1	顧客付款: <u>100</u>
蛋糕(\$12) 2	找回款項: \$52
蘋果(\$5) 3	
西瓜(\$30) 0	

14. 宣告整數陣列: mark[] (40 個元素)

產生 40 個隨機數 (0-100)

顯示分數 (每行 5 個)

若 mark[n] < 50, 則 +10 分, 再顯示分數 (每行 5 個)

顯示總分、平均分、最高、最低分數、合格人數

41	85	72	38	80	加分後				
69	65	68	96	22	51	85	72	48	80
...					69	65	68	96	32
總分 = 2302	平均分 = 57.5				...				
合格人數 = 26									
最高分數 = 99	最低分數 = 11								

14. 宣告整數陣列：mark[] (40 個元素) 及 freq[] (10 個元素)

輸入 40 個分數 (0-100)

顯示頻數分佈表 frequency distribution table

```

Q: 輸入分數 (0-100) : 999
輸入錯誤
Q: 輸入分數 (0-100) : 99
Q: 輸入分數 (0-100) : 10
...

```

顯示結果:

分數	頻數		
00-09	0	(0.0%)	
10-19	1	(2.5%)	*
20-29	3	(7.5%)	***
30-39	5	(12.5%)	*****
40-49	3	(7.5%)	***
50-59	11	(27.5%)	*****
60-69	9	(22.5%)	*****
70-79	5	(12.5%)	*****
80-89	2	(5.0%)	**
90-100	1	(2.5%)	*
總人數	40	(100.0%)	

15. 多項選擇題 MC Questions & Answers

answers[]

A	A	D	C	B	...				
---	---	---	---	---	-----	--	--	--	--

執行畫面如下：

Q 1: A	Q 1: A
Q 2: B	Q 2: C
...	...
Q10: D	Q10: A
A: 答對 7 題，70 分。	A: 答對 3 題，30 分。
Q1-10? ABCDEABCDE	
A: 答對 5 題，50 分。	

16. 宣告陣列 box[] (盒)，每個盒的起始值為 2，排列成一個圓形。

重覆以下步驟：

產生隨機數 n ($0 \leq n \leq 5$)把第 n 個盒的值加 2

並把相鄰的 2 個盒的值各減 1

直至任何一個盒的數值為 0

例如：

		box[]					
	n	[0]	[1]	[2]	[3]	[4]	[5]
開始前	-	2	2	2	2	2	2
第 1 回	2	2	1	4	1	2	2
第 2 回	0	4	0	4	1	2	1

執行畫面如下：

```

box[]  0  1  2  3  4  5
n=2    2  1  4  1  2  2
n=0    4  0  4  1  2  1
Game Over

```

變化 1：

只要任何 2 個數值相差多於 5，便停止

變化 2：

1.宣告整數陣列: box[] (10 個元素)

2.box[0]=5; ...

3.n=random(0-9)

nth box +2

若 n 是單數,其餘所有單數 box +1

所有雙數 box -1

否則 其餘所有雙數 box +1

所有單數 box -1

4.重複 3,直至出現 0 或 10

5.顯示哪個 box 出現 0 或 10

box[4]=0 或 box[7]=10

17. 擲骰子(1 或 2 粒) Throw a die (1-6) / 2 dice (2-12)

outcome[]	1	2	3	4	5	6
freq[]	200	190	210	180	220	200
percent[]	16%					

n = rand()%?+?;

<p>17. 擲骰子 rolling-a-die</p> <p>Q: How many time do you want to roll a die? 12000</p> <p>outcome 1: 2000 (17%)</p> <p>outcome 2: 1980 (15%)</p> <p>...</p> <p>outcome 6: 2010 (18%)</p>	<p>17. 擲骰子 rolling-2-dice (12000 times)</p> <p>outcome 2: 300 (2.5%)</p> <p>...</p> <p>outcome 12: 400 (3.3%)</p>
---	---

18. 分數 Score & 等級 Grade (40 人)

score[]	40	50	70	80	90	...				
grade[]	E	D	C	B	A	...				

執行畫面如下：

輸入	輸出
學生 1: 40	學生 1: 40 E
學生 2: 50	學生 2: 50 D
...	...
學生 40: 70	A(3) B(7) C(10) D(15) E(5)

19. 分數統計

- 輸入分數 或 按次增加 (e.g. 30,35,40,...)
- 產生 40 個隨機數 (30-100)，儲存於陣列 mark[] (數值可能重複)
- 顯示分數：每行 5 個分數 (共 8 行)
- 計算及顯示總分、平均分、合格人數
- 凡不合格者，各加 10 分；少於 80 分者，各加 5 分
- 分數搜尋：100 (使用者輸入數值 v)

輸入分數：100 學生 10 得 100 分	輸入分數：100 得 100 分學生人數：3 人
輸入分數：90 找不到	輸入分數：90 得 90 分學生人數：0 人

- 利用 qsort 排序 + binary search

20. 水費單 Water Bill: 住宅用水計算方法 (耗水量 m^3 及收費)

分段(m^3) interval[]	首 12	其次 31	其次 19	其餘(m^3)
每立方米收費 rate[]	\$0.00	\$4.16	\$6.45	\$9.05

執行畫面如下：

Q: 耗水量(立方米)? 12	A: 收費 = \$0.00
Q: 耗水量(立方米)? 22	A: 收費 = $0.00 \times 12 + 4.16 \times 10 = \41.6
Q: 耗水量(立方米)? 53	A: 收費 = $0.00 \times 12 + 4.16 \times 31 + 6.45 \times 10 = \193.5
Q: 耗水量(立方米)? 72	A: 收費 = $0.00 \times 12 + 4.16 \times 31 + 6.45 \times 19 + 9.05 \times 10 = \342.0

已知: int consumption; // 耗水量 m^3
試寫 C 程式，利用陣列，計算應收水費。

21. 撲克牌遊戲 Blackjack:

- 宣告(撲克牌)陣列 `card[]`，用作儲存隨機數 0-51
- 產生 52 個隨機數 (0-51)，儲存於陣列 `card[]` (數值不可重複)

例如：

```
card[]  28  25  50  0  ...  ...  ...  ...  ...
```

提示：利用另一陣列 `num[]`，記著每個數值(0-51)是否已經出現。

```
num[]   1   1   0   1  ...  ...  ...  ...  ...
```

本程式與六合彩程式類同。

- 根據點數對照表，顯示所有撲克牌及點數：

第 1 張：28 (梅花 3) 點數：3

第 2 張：25 (紅心 K) 點數：10

第 3 張：50 (階磚 Q) 點數：10

點數對照表：

點數	1	2	3	4	5	6	7	8	9	10	J	Q	K	
儲	♠	0	1	2	3	4	5	6	7	8	9	10	11	12
存	♥	13	14	15	16	17	18	19	20	21	22	23	24	25
數	♣	26	27	28	29	30	31	32	33	34	35	36	37	38
值	♦	39	40	41	42	43	44	45	46	47	48	49	50	51

花色的等級： 葵扇♠ > 紅心♥ > 梅花♣ > 階磚♦。

- 依次序派牌給電腦及玩者，只顯示玩者點數，不顯示電腦點數，直至有一方點數達到或超過 21。
- 決定誰勝誰負。

22. 洗牌 Shuffle Card (1-52)

```
card[]  20  11  43  25  51  7  31  ...  ...
```

```
c = a; a = b; b = c; // 互換 swapping a & b
```

```
x = rand()%52+1;
```

```
y = rand()%52+1;
```

```
// 互換 swapping card[x] & card[y]
```

23a. 質數 Prime Number (1-50)

prime[01]	-	0	1	0	1	0	1	0	1	0
prime[11]	1	0	1	0	1	0	1	0	1	0
prime[21]	1	0	1	0	1	0	1	0	1	0
prime[31]	1	0	1	0	1	0	1	0	1	0
prime[41]	1	0	1	0	1	0	1	0	1	0

0. 初始化陣列 `prime[i]`

```
prime[i]  -  1  1  1  1  1  1  1  1  ...
```

1. 要求使用者輸入 n ($n > 1$)
2. 把 n 的所有倍數刪除 (除了 n 本身以外), 例如: `prime[2*n]=0`
3. 把餘下未被刪除的數字顯示在屏幕上
4. 重覆步驟 1-3

Q: 輸入 n ($n > 1$) ? 2
 A: 餘下的數字: 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, ...

Q: 輸入 n ($n > 1$) ? 3
 A: 餘下的數字: 2, 3, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, ...

Q: 輸入 n ($n > 1$) ? 5
 A: 餘下的數字: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47

Q: 輸入 n ($n > 1$) ? 7
 A: 餘下的數字: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47

23b. 質數 Prime Number (1-20)

```
#define MAX 50
int prime[MAX];
for(i=1; i<=MAX; i++) prime[i]=1; // assume 假設 all prime
prime[1]=0; // 1 不是質數
```

d*i	2	3	4	5	6	7	8	9	10
d=2	4	6	8	10	12	14	16	18	20
d=3	6	9	12	15	18				
d=4	8	12	16	20					
5	10	15	20						
6	12	18							
7	14	21							
8	16								
9	18								
10	20								

由 $d=2$ 開始, 把 d 的倍數 ($d*1$ 除外) 抽出標示為非質數。例如:

```
d=2; prime[4]=0; prime[6]=0; prime[8]=0; ...
d=3; prime[6]=0; prime[9]=0; prime[12]=0; ...
d=4; prime[8]=0; prime[12]=0; prime[16]=0; ...
```

```
d=2;
do{
    for (i=2; i<=??; i++) prime[??]=1;
    d++;
```

```
}while(d<=??);
```

如何優化(加快)以上程式?

24. 六合彩 Mark VI (1-49)

```
num[]  0  1  1  1  0  1  0  ...  ...
#define MAX 49
int num[MAX+1];
for(n=1;n<=MAX;n++) num[n]=0;    // 數字 n 未被抽中

for(i=1;...){
    n = rand()%...
    num[n] = 1;    // 數字 n 已被抽中
}
```

如何解決 1 個數字被抽中 2 次或以上? (利用 do..while)

25. Lunch Box (A,B,C 餐) - frequency table

```
student[]  A  C  B  A  A  C  C  ...  ...
lunchbox[] 12  8  17  ...  ...

char student[40];
int lunchbox[3];
for(i=0;i<MAX;i++){
    n=rand()%3;    // 0-2
    student[i]=n+'A';
}
```

26. 薪俸稅 Salary Tax: http://www.ird.gov.hk/chi/tax/ind_tra.htm

	應課稅入息實額 \$s	累進稅率 r%	稅款 \$tax
首	35,000	2%	700
另	35,000	7%	2,450
另	35,000	12%	4,200
餘額		17%	

執行畫面如下：標準稅率=16%

```
Q: 應課稅入息實額 $s? 10000
A: 稅款 = 10000×2% = $200
Q: 應課稅入息實額 $s? 45000
A: 稅款 = 35000×2% + 10000×7% = $1400
Q: 應課稅入息實額 $s? 80000
A: 稅款 = 35000×2% + 35000×7% + 10000×12% = $4350
```

27. Queue / Stack / Linked List

queue[]	12	8	17						
---------	----	---	----	--	--	--	--	--	--

執行畫面如下：

<E> EnQueue +	Option? D
<D> DeQueue -	12 deleted from Queue.
<L> List Queue	Option? L
	8 < 17
Option? E	Option? E
EnQueue? 12 8 17 0	EnQueue? 12 99 0
Option? L	Option? L
12 < 8 < 17	8 < 17 < 12 < 99

28. Table assignment (max 10/table)

table[]	9	5	3	0					
---------	---	---	---	---	--	--	--	--	--

執行畫面如下：

Q: No.of people? 5	Q: No.of people? 7
Smoker? N	Smoker? Y
Sit together? Y	Sit together? N
A: Table 2	A: Table 1 & 3

Table 1,3,5,... (Smoker), Table 2,4,6,... (Non-smoker)

29. Heat 組 & Lane 線道:

int laneArray[8]={4,5,3,6,2,7,1,8};

heat[50]	1	1	1	1	1	1	1	1	2	2
lane[50]	4	5	3	6	2	7	1	8	4	5

for(i=0;i<50;i++){

heat[i] = ?

lane[i] = ?

}

30. 選舉 Ballot (4 candidates 候選人 / 0 void 廢票)

選票 vote[]	1	1	3	1	2	4	0	1	2	...
頻數 freq[]	22	15	32	10						

for(i=0;i<1000;i++) vote[i] = rand()%5;

執行畫面如下：

Candidate 1	22	...
Candidate 2	15	Candidate 3 wins.
Candidate 3	32	
Candidate 4	10	