

《程式設計概要》

試題評析

今年考出題型多與往年相同，包含三題的C語言程式與一題的Javascript程式。

第一題使用迴圈加上陣列就可以解決。

第二題的鏈結串列問題考的是雙重指標，可以考出考生程度。

第三題搜尋子字串是普通的迴圈問題。

第四題的Javascript考題使用window物件與document物件控制網頁頁面的文字輸入方塊。

程度好的考生應可獲得80分以上。

一、用C語言撰寫一個函式int most_significant_set_bit(int num)，它能將一個32-bit整數的最高有效位 (The most significant bit) 為1的位置計算出並回傳出來，如果全為0，則回傳-1。例如數值444以二進位表示為 00000000 00000000 00000001 10111100，所以它回傳出最高有效位為1的位置為8，也就是說bit8。(25分)

答：

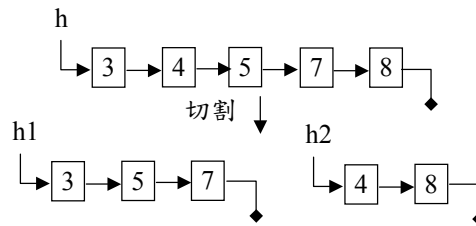
```
#include <stdio.h>
// 本函數只處理無號數
int most_significant_set_bit(int num){
    int binary[32];
    int i;
    // input error
    if(num<0)
        return(-2);
    // 將陣列清為0
    for(i=0; i<32; i++)
        binary[i]=0;
    // 將num轉二進位後存入陣列binary
    i=0;
    while(num!=0){
        binary[i]=num%2;
        num/=2;
        i++;
    }
    for(i=0; i<32; i++)
        printf("%d ", binary[i]);
    printf("\n");

    // 找出第一個出現bit 1的位置
    for(i=31; i>=0; i--)
        if(binary[i]==1)
            return(i);
    return(-1);
}
int main(void){
    printf("most bit: %d\n", most_significant_set_bit(444));
}
```

二、用C語言撰寫一個函式，能將一只含有數字的單向鏈結串列 (singly linked list) 切割成兩個單向鏈結串列，其一只包含奇數元素，另一只包含偶數元素，請勿複製節點，切割前後都是以數字由小到大排序，假設此函式的原型 (prototype) 如下：void split (node *h, node **h1, node **h2)，h是切割前鏈結串列兩個單向鏈結串列指標，h1, h2是切割後兩個單向鏈結串列的指標。(25分)

其中節點的資料結構為

```
typedef struct node {
    int d;
    struct node *next;
}node;
```



單向鏈結串列範例

答：

```
#include <stdio.h>
typedef struct node {
    int d;
    struct node *next;
}node;
void printList(node *p){
    while(p!=NULL){
        printf("%d ", p->d);
        p=p->next;
    }
    printf("\n");
}
void split(node *h, node **h1, node **h2){
    node *tail1=NULL, *tail2=NULL;
    while(h!=NULL){
        if((h->d)%2==0){
            if(*h1==NULL){
                *h1=h;
                h=h->next;
                tail1=*h1;
                tail1->next=NULL;
            }else{
                tail1->next=h;
                h=h->next;
                tail1=tail1->next;
                tail1->next=NULL;
            }
        }else{
            if(*h2==NULL){
                *h2=h;
                h=h->next;
                tail2=*h2;
                tail2->next=NULL;
            }else{
                tail2->next=h;
                h=h->next;
                tail2=tail2->next;
                tail2->next=NULL;
            }
        }
    }
}
```

```

    }
}
int main(void){
    node *h=NULL, *h1=NULL, *h2=NULL;
    node n3={3, NULL}, n4={4, NULL}, n5={5, NULL}, n7={7, NULL}, n8={8, NULL};
    h=&n3; n3.next=&n4; n4.next=&n5; n5.next=&n7; n7.next=&n8;
    printList(h);
    split(h, &h1, &h2);
    printList(h1);
    printList(h2);
}

```

三、用任何高階電腦語言撰寫一個函式 `int string_search(char str[], char pat[])`，它能計算出子字串 `pat` 在字串 `str` 中的位置。例如，假設 `str="I will pass this exam."` 且 `pat="pass"`，則函式 `string_search(str, pat)` 回傳 7。（25分）

答：

```

#include <stdio.h>
#include <string.h>
int string_search(char str[], char pat[]){
    int i,j,pLen,cLen;
    pLen=strlen(str);
    cLen=strlen(pat);
    for(i=0; i<=pLen-cLen; i++){
        for(j=0; j<cLen; j++){
            if(str[i+j]!=pat[j]) break;
        }
        if(j==cLen) return(i);
    }
    return(-1);
}
int main(void){
    char str[]="I will pass this exam.";
    char pat[]="pass";

    printf("str: %s\n", str);
    printf("pat: %s\n", pat);
    printf("%d\n", string_search(str, pat));
}

```

四、利用 Javascript 之 `onClick` 功能來製作如圖(a)讓使用者可由表單 (form) 輸入四個 `inputs`，第一個輸入是 server 端程式的 `url` 字串，第二三四個輸入是三個 `keys` 的值，用以組合 `query string`。待 Javascript 程式組合好包含 `url` 及 `query string` 的 `request` 字串後，再藉由 `submit button` 點擊 (click) 後傳送出去。例如，如果 `url` 輸入的是 `"wwwc.moex.gov.tw/lp.asp"`，三個 `keys` 的值分別為 1, 2, 3，則所組合好的 `request` 字串則為 `http://wwwc.moex.gov.tw/lp.asp?key1=1&key2=2&key3=3`，當 `submit button` 被點擊 (click) 後，則以 `request` 字串為名稱的網頁就會抓回來。請將圖(b)空格內之編號及答案依序寫在試卷上。（25分，其中(7)、(21)、(22)各為2分，其餘均為1分）

Query String Example

url:

key1:

key2:

key3:

圖(a)

```

<Script Language=JavaScript>
function Location() {
  var UrlStr;
  UrlStr="http://" + document.LocationBody.url.value +
    (1) + (2) +
    (3) + (4) +
    (5) + (6) ;
  window.location.href= (7) ;
}
</Script>
<Form Name=LocationBody>
(8) <Input Type= (9) Name= (10) ><BR>
(11) <Input Type= (12) Name= (13) ><BR>
(14) <Input Type= (15) Name= (16) ><BR>
(17) <Input Type= (18) Name= (19) ><BR>
<Input Type= (20) Value= (21) onClick= (22) >
</Form>

</BODY></HTML>

```

圖(b)

答：

```

<html>
<body>
<script type="text/javascript">
function Location(){
  var UrlStr;
  UrlStr="http://" + document.LocationBody.url.value +
  "?key1=" + document.LocationBody.key1.value +
  "&key2=" + document.LocationBody.key2.value +
  "&key3=" + document.LocationBody.key3.value
  ;
  window.location.href=UrlStr;

```

```
}  
</Script>  
<Form Name=LocationBody>  
url:<Input Type='text' Name='url'><BR>  
key1:<Input Type='text' Name='key1'><BR>  
key2:<Input Type='text' Name='key2'><BR>  
key3:<Input Type='text' Name='key3'><BR>  
<Input Type='button' Value='submit' onClick='Location()'>  
</Form>  
  
</Body></Html>
```

【參考書目】

1. 胡世雄，《程式設計概要》，高點文化出版。

