Data Structure

姓名：朱鈺劭　　學號：CM100124　　(Due: 2012/4/10)

1.設有一陣列 A[-1:3, 2:4, 1:4]以列為主(row major)儲存資料，陣列 A 的起位址為 100， 每個元素需要 2 個記憶空間，則 A[3,3,3]位址為何?

Ans:
A[u1, u2, u3] = A[-1:3, 2:4, 1:4] → u1 = 5, u2 = 3, u3 = 4。
A[3, 3, 3] →表示 A[-1:3, 2, 1]的範圍都列入計算，因此 4*u2*u3*2 = 96，
A[3, 2, 1]之記憶體位置為 100+96 = 196。
再往繼續推算 A[3, 2:3, 1]的範圍亦列入計算，因此 1*u3*2 = 8，
A[3, 3, 1]之記憶體位置為 196+8 = 204。
最後，再將 A[3, 3, 1:3]的範圍列入計算，2*2 = 4，
可得 A[3, 3, 3]之記憶體位置為 204+4 = 208。

2 利用堆疊方法將中序式 A+B*C/(E-2)*F 轉成後序式

Ans:	Description	in-Stack	output
Step1.	A 為運算元，直接輸出		A
Step2.	'+' 直接推入空堆疊	+	A
Step3.	B 為運算元，直接輸出	+	AB
Step4.	'*' 之 ICP > '+' 之 ISP	*+	AB
Step5.	C 為運算元，直接輸出	*+	ABC
Step6.	'/' 之 ICP = '*' 之 ISP，POP '*'，之後再推入 '/'	/+	ABC*
Step7.	'(' 之 ICP > '/' 之 ISP	(/+	ABC*
Step8.	E 為運算元，直接輸出	(/+	ABC*E
Step9.	'-' 之 ICP > '(' 之 ISP	-(/+	ABC*E
Step10.	2 為運算元，直接輸出	-(/+	ABC*E2
Step11.	遇到 ')'，不斷 POP 直到遇到 ')'，但不輸出 '(' 與 ')'	/+	ABC*E2-
Step12.	'*' 之 ICP = '/' 之 ISP，POP '/'，之後再推入 '*'	*+	ABC*E2-/
Step13.	F 為運算元，直接輸出	*+	ABC*E2-/F
Step14.	None，POP 所有堆疊內容		ABC*E2-/F*+

3. 寫程式利用雙向鏈節串列模擬堆疊，包含 push, pop 與顯示堆疊內容

Ans:

```
/*  Author:  CM100124
    Title:   Data Stucture HW#1
    Date:    2012/4/2              */
#include <stdio.h>
#include <stdlib.h>

void push(void);
void pop(void);
void show_stack(void);

struct stack {
      char data[10];
      struct stack *llink;
      struct stack *rlink;
      };

struct stack *bottom,*head,*ptr,*top,*prev,*read;

int main(){
   head = (struct stack *) malloc(sizeof(struct stack));
   strcpy(head->data ,"Head");
   bottom = (struct stack *) malloc(sizeof(struct stack));
   strcpy(bottom->data ,"End");
   head->rlink = bottom;
   head->llink = bottom;
   bottom->llink = head;
   bottom->rlink = head;
   top = bottom;
   prev = bottom;
   printf("Program Initialized.\n");

   char option;

   while(1){
   printf("I want to:\n 1.Push\n 2.Pop\n 3.Show Stack\n 4.Exit");
```

```
  option = getch();
  switch(option){
    case '1':
       push();
       break;
    case '2':
       pop();
       break;
    case '3':
       show_stack();
       break;
    case '4':
       return 0;
       break;
  }
  }
  system("pause");
  return 0;
  }

void push(void){
   printf("\n\n");
   ptr = (struct stack *) malloc(sizeof(struct stack));
   printf("Enter the string you want to push(length of string<10):");
   gets(ptr->data);
   prev = head->rlink;
   head->rlink = ptr;
   top->llink = ptr;
   ptr->llink = head;
   ptr->rlink = top;
   top = ptr;
   printf("Push Success!\n\n");
   return;
   }

 void pop(void){
   printf("\n\n");
   if(strcmp(top->data,"End")==0)
```

```
printf("This is the bottom of the stack, please push a new value.\n\n");
else{
printf("Pop out: %s\n\n",top->data);
head->rlink = top->rlink;
prev->llink = head;
free(top);
top = prev;
prev = prev->rlink;
}
return;
}

void show_stack(void){
   printf("\n\nShow the whole stack:\n");
   read = head->rlink;
   while(strcmp(read->data,"End")!=0){
     printf("%s\n",read->data);
     read = read->rlink;
   }
   printf("\n");
   return;
   }
```

**********(Execute Program)**********

```
I want to:
1.Push
2.Pop
3.Show Stack
4.Exit
→1

Enter the string you want to push(length of string<10):
→Eric

I want to:
1.Push
2.Pop
3.Show Stack
```

4.Exit
→1

Enter the string you want to push(length of string<10):
→Tiffany

I want to:
1.Push
2.Pop
3.Show Stack
4.Exit
→1

Enter the string you want to push(length of string<10):
→Nick

I want to:
1.Push
2.Pop
3.Show Stack
4.Exit
→2

Pop out:Nick

I want to:
1.Push
2.Pop
3.Show Stack
4.Exit
→3

Show the whole stack:
Tiffany
Eric

**********(End)**********

4. Ackerman(m,n)=n+1 if m=0
Ackerman(m,n)= Ackerman(m-1,1) if m>0,n=0
Ackerman(m,n)= Ackerman(m-1, Ackerman(m,n-1)) if m>0,n>0
(a) A(1,3)=?
(b)請利用遞迴方式完成此程式

Ans:

```
#include <stdlib.h>
int main(void){
   printf("Ackerman(1,3)= %d\n",Ackerman(1,3));
   system("pause");
   return 0; }
int Ackerman(int m, int n){
    if(m==0)
    return n+1;
    if(m>0 && n==0)
    return Ackerman(m-1,1);
    if(m>0 && n>0)
    return Ackerman(m-1, Ackerman(m,n-1));}
```

**********(Execute Program)**********

Ackerman(1,3) = 5

**********(End)**********

5. 假設 n 為 2 的次方，試問何者為下列程式片段的時間複雜度
(a) n (b) $(\log n)^2$ (c) n*log(n) (d) n^2

Ans: (The log is to the base 2)

	Code	Times
1.	i=n;	1
2.	while (i >=1)	log(n)+2
3.	{	
4.	j=i;	log(n)+1
5.	while (j <= n)	[4+log(n)]*[log(n)+1]/2
6.	{ j=2*j; }	[2+log(n)]*[log(n)+1]/2
7.	i=i/2;	log(n)+1
8.	}	

由上可得知，程式之執行時間為：$(\log n)^2 + 7\log(n) + 8$，時間複雜度為 $(\log n)^2$ 答案為(b)。