

CP 111 Computer Programming, 2/2554, Section B01/B02

Programming Homework 08

กำหนดส่งภายในวันจันทร์ที่ 20 ก.พ. 2555 เวลา 23.59 น.

คำเตือน

นิสิตสามารถจับกลุ่มปรึกษาหารือกันได้ แต่นิสิตแต่ละคนในกลุ่มจะต้องลงมือเขียนโปรแกรมด้วยตนเองและส่งโปรแกรมของตนเองโดยไม่ขึ้นอยู่กับสมาชิกในกลุ่ม และอย่านำโปรแกรมของตนเองให้กับผู้อื่นดู
ถ้าตรวจพบว่านิสิตนำงานของคนอื่นมาส่งหรือนิสิตก๊อปปี้ แก้ไข คัดแปลงโปรแกรมของผู้อื่น จะมีโทษคือปรับตก (ได้เกรด E) ในวิชานี้ทันที

หมายเหตุ

- ข้อกำหนดการตั้งชื่อไฟล์การบ้านคือ xxx_HWyy_Qzz.* โดยที่ xxx = รหัสสามตัวท้ายของเลขประจำตัว, yy = หมายเลข homework, zz = หมายเลขข้อในแต่ละ homework (ตัวอย่างเช่น ไฟล์โปรแกรม ข้อ 1 การบ้านครั้งที่ 1 ของนิสิตรหัส 000 คือ 000_HW01_Q01.c) และในแต่ละข้อ
- การส่งการบ้าน ให้ส่งไฟล์โปรแกรมไปตรวจกับระบบ Café-grader ที่ <http://cslab.swu.ac.th/grader>
- การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

1. Problem name: Selection_Sort

```
#include <stdio.h>
void SelectionSort(int *A,int n,int order);
int main()
{
    int data[100];
    int tmp,N,i,j;

    scanf("%d",&N);
    for(i=0;i<N;i++)
        scanf("%d",&data[i]);

    SelectionSort(data,N,1);

    for(i=0;i<N;i++)
        printf("%d\n",data[i]);
    return 0;
}

void SelectionSort(int *A,int n,int order)
{
    int i,j,m_index,m_val,tmp;

    for(i=0;i<n-1;i++)
    {
        // find min among A[i] ... A[n-1]
        m_val=A[i];
        m_index=i;
        for(j=i;j<n;j++)
        {
            if(order==0)
            {
                if(A[j]<m_val)
                {
                    m_val=A[j];
                    m_index=j;
                }
            }
            else
            {
                if(A[j]>m_val)
                {
                    m_val=A[j];
                    m_index=j;
                }
            }
        }
        // Swap A[i] <-> A[min_index]
        tmp=A[i];
        A[i]=A[m_index];
        A[m_index]=tmp;
    }
}
```

2. Problem name: What_in_parentheses

```
int main()
{
char instr[501];
int i,k,state,open_p,close_p;

    gets(instr);

    i=0;
    state=0;
    while(1)
    {
        if(state==0)
        {
            if(instr[i]=='(' )
            {
                state=1;
                open_p=i;
            }
        }
        else if(state==1)
        {
            if(instr[i]==')')
            {
                close_p=i;
                printf("%d ",close_p-1-(open_p+1)+1);
                for(k=open_p+1;k<=close_p-1;k++)
                    printf("%c",instr[k]);
                printf("\n");
                state=0;
            }
        }
        if(instr[i]==0)
            break;
        i=i+1;
    }
    return 0;
}
```

3. Problem name: Max_consecutive_sum

```
#include <stdio.h>

void Max_Consecutive_Sum(int *A,int N, int *max_sum,int *first, int *last);

int main()
{
    int X[100];
    int N,Nk,n_cons,i,j,max_sum,max,sum,n_cases,first,last;

    scanf("%d",&N);
    for(i=0;i<N;i++)
        scanf("%d",&X[i]);

    Max_Consecutive_Sum(X,N,&max_sum,&first,&last);
    printf("%d\n",max_sum);

    return 0;
}

void Max_Consecutive_Sum(int *A,int N, int *max_sum,int *first, int *last)
{
    int i,j,nC,sum;

    for(i=0;i<N;i++)
    {
        if(i==0)
        {
            *max_sum=A[i];
            *first=i;
            *last=i;
        }

        for(nC=0;nC<N-i;nC++)
        {
            sum=0;
            for(j=0;j<=nC;j++)
            {
                sum=sum+A[i+j];
                if(sum>*max_sum)
                {
                    *max_sum=sum;
                    *first=i;
                    *last=j;
                }
            }
        }
    }
}
```

4. Problem name: Reverse_word

```
#include <stdio.h>
#define MAX_STR_LEN 500
int main()
{
    char instr[MAX_STR_LEN+1];
    char outstr[MAX_STR_LEN+1];
    int count,i,j,k,len,first,last,state;

    gets(instr);
    state=0;
    i=0;
    while(instr[i]!=0)
    {
        if(state==0 )
        {
            if(instr[i]!=' ')
            {
                first=i;
                state=1;
            }
            else
            {
                first=-1;
                outstr[i]=instr[i];
            }
        }
        else if(state==1)
        {
            if(instr[i]==' ')
            {
                last=i-1;
                state=0;
                for(k=last,j=first;k>=first;k--)
                {
                    outstr[j]=instr[k];
                    j++;
                }
                outstr[j]=' ';
            }
        }
        i++;
    }
    if(state==1)
    {
        last=i-1;
        state=0;
        for(k=last,j=first;k>=first;k--)
        {
            outstr[j]=instr[k];
            j++;
        }
        outstr[j]=0;
    }
    else
        outstr[j]=0;

    printf("%s\n",outstr);

    return 0;
}
```

5. Problem name: Substring

```
#include <stdio.h>

#define MAX_STR_LEN 500

int string_length(char *str);
int substring(char *instr, char *t_str);

int main()
{
    char instr[MAX_STR_LEN+1];
    char t_str[MAX_STR_LEN+1];
    int count, i, k, len, t_len, match;

    gets(instr);
    gets(t_str);
    match=substring(instr, t_str);
    printf("%d\n", match);
}

int substring(char *instr, char *t_str)
{
    int i, k, match;

    i=0;
    while(instr[i]!=0)
    {
        k=0;
        match=1;
        while(t_str[k]!=0)
        {
            if(instr[i+k]!=t_str[k])
            {
                match=0;
                break;
            }
            k=k+1;
        }
        if(match==1)
            return(i);

        i=i+1;
    }
    return -1;
}
```

6. Problem name: *Matrix_Transpose*

```
#include <stdio.h>
#define MAX_N 20

int main()
{
    int A[MAX_N][MAX_N];
    int At[MAX_N][MAX_N];
    int C[MAX_N][MAX_N];
    int M,N,i,j,k;

    scanf("%d %d" ,&M,&N);
    for(i=0;i<M;i++)
        for(j=0;j<N;j++)
            scanf("%d",&A[i][j]);

    for(i=0;i<N;i++)
        for(j=0;j<M;j++)
            At[i][j]=A[j][i];

    for(i=0;i<N;i++)
        for(j=0;j<N;j++)
        {
            C[i][j]=0;
            for(k=0;k<M;k++)
                C[i][j]=C[i][j]+(At[i][k]*A[k][j]);
        }

    for(i=0;i<N;i++){
        for(j=0;j<N;j++)
            printf("%d ",C[i][j]);
        printf("\n");
    }

    return 0;
}
```

7. Problem name: Straight (ไผ่ตอง)

```
void SelectionSort(int *A,int n);
int main()
{
    int i,j,k,X[10];
    int X1,X2,X3,found;
    for(i=0;i<10;i++)
    {
        scanf("%d",&X[i]);
    }

    SelectionSort(X,10);

    found=0;
    for(i=0;i<10;i++)
    {
        for(j=i+1;j<10;j++)
        {
            for(k=j+1;k<10;k++)
            {
                if(X[i]+1==X[j] && X[j]+1 == X[k])
                {
                    found=1;
                    X1=X[i];
                    X2=X[j];
                    X3=X[k];
                }
            }
        }
    }
    if(found!=0)
    {
        printf("%d %d %d\n",X1,X2,X3);
    }
    else
    {
        printf("-1\n");
    }
    return 0;
}

void SelectionSort(int *A,int n)
{
    int i,j,min_index,min_val,tmp;
    for(i=0;i<n-1;i++)
    {
        // find min among A[i] ... A[n-1]
        min_val=A[i];
        min_index=i;
        for(j=i;j<n;j++)
        {
            if(A[j]<min_val)
            {
                min_val=A[j];
                min_index=j;
            }
        }
        // Swap A[i] <-> A[min_index]
        tmp=A[i];
        A[i]=A[min_index];
        A[min_index]=tmp;
    }
}
```