

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

Programming Homework 10

1. Problem name: Extract_URL_stdin

```
#include <stdio.h>
#define MAX_STR_LEN 500
int main()
{
    char URL[MAX_STR_LEN+1], ch;
    int N, state;

    N=0;
    state=0;
    while(1)
    {
        ch=fgetc(stdin);
        if(ch=='\n')
            break;

        if(state==0)
        {
            if(ch=='')
                state=1;
        }
        else if(state==1)
        {
            if(ch=='h');
            state=2;
        }
        else if(state==2)
        {
            if(ch=='t');
            state=3;
        }
        else if(state==3)
        {
            if(ch=='t');
            state=4;
        }
        else if(state==4)
        {
            if(ch=='p');
            state=5;
        }
        else if(state==5)
        {
            if(ch==':');
            state=6;
        }
        else if(state==6)
        {
            if(ch=='/');
            state=7;
        }
        else if(state==7)
        {

```

```

        if(ch=='/');
        {
            state=8;
        }
    }
    else if(state==8)
    {
        if(ch!='"')
        {
            URL[N]=ch;
            N++;
        }
        else
        {
            URL[N]=0;
            printf("\nhttp://%s"\n",URL);
            N=0;
            state=0;
        }
    }
}

return 0;
}

```

2. Problem name: Find_students

```

#include <stdio.h>
#define MAX_DATA 50

typedef struct
{
    int id;
    char name[31];
    char gender;
    char school[31];
    int year;
    int score;
} S_INFO;

int main()
{
    S_INFO s[MAX_DATA];
    int N,i;
    double avg;
    scanf("%d",&N);
    for(i=0;i<N;i++)
    {
        scanf("%d %s %c %s %d %d",&(s[i].id),s[i].name,&(s[i].gender),
            s[i].school,&(s[i].year),&(s[i].score));
    }

    // Find the average score
    avg=0;
    for(i=0;i<N;i++)
    {
        avg=avg+(double) (s[i].score);
    }
    avg=avg/N;

    // Display the result

```

```

printf("%.2f\n", avg);
for(i=0; i<N; i++)
{
    if(s[i].gender=='M' && s[i].year==1 && s[i].score >=avg)
    {
        printf("%d %s %c %s %d %d\n", s[i].id, s[i].name, s[i].gender,
            s[i].school, s[i].year, s[i].score);
    }
}

return 0;
}

```

3. Problem name: Sort_GPA_student

```

#include <stdio.h>
#define MAX_N 50
typedef struct {
    int id;
    char fname[21];
    char lname[51];
    char gender;
    int age;
    double gpa;
} STUDENT_INFO;
int main()
{
    STUDENT_INFO s[MAX_N];
    STUDENT_INFO tmp_s;
    int i, j, N;
    scanf("%d", &N);
    for(i=0; i<N; i++)
    {
        scanf("%d %s %s %c %d %lf", &(s[i].id), s[i].fname,
            s[i].lname, &(s[i].gender),
            &(s[i].age), &(s[i].gpa));
    }
    for(i=0; i<=N-2; i++)
    {
        for(j=N-1; j>=i+1; j--)
        {
            if(s[j].gpa > s[j-1].gpa )
            {
                tmp_s=s[j];
                s[j]=s[j-1];
                s[j-1]=tmp_s;
            }
        }
    }
    for(i=0; i<N; i++)
    {
        printf("%d %s %s %c %d %.2f\n", s[i].id,
            s[i].fname,
            s[i].lname,
            s[i].gender,
            s[i].age,
            s[i].gpa);
    }

    return 0;
}

```

4. Problem name: Closest_pairs_of_3D_points

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
typedef struct{
    int id;
    double x;
    double y;
    double z;
} Point3D;
Point3D* Find_Closest_Point(Point3D *point_set,
                           int N_points, Point3D *target_point);

int main()
{
    int i, N;
    Point3D *pts;
    Point3D *closest_pts;

    scanf("%d", &N);

    pts=(Point3D *)malloc(N*sizeof(Point3D));
    if(pts==NULL){
        printf("Memory Allocation ERROR\n");
        return -1;
    }

    for(i=0; i<N; i++)
    {
        scanf("%d %lf %lf %lf", &(pts[i].id),
              &(pts[i].x),
              &(pts[i].y),
              &(pts[i].z));
    }
    for(i=0; i<N; i++)
    {
        closest_pts=Find_Closest_Point(pts, N, &pts[i]);
        printf("%d %d\n", pts[i].id, closest_pts->id);
    }

    free(pts);

    return 0;
}

Point3D* Find_Closest_Point(Point3D *point_set, int N_points,
                           Point3D *target_point)
{
    int i;
    int min_i;
    double d, min_d;
    double x1, y1, z1, x2, y2, z2;

    min_i=-1;
    min_d=-1;
    for(i=0; i<N_points; i++){
        if(target_point->id == point_set[i].id)
            continue;
        x1=point_set[i].x;
        y1=point_set[i].y;
        z1=point_set[i].z;
        x2=target_point->x;
```

```

        y2=target_point->y;
        z2=target_point->z;
        d = sqrt( pow(x1-x2,2) + pow(y1-y2,2) + pow(z1-z2,2) );

        if(min_d==-1 || (min_d!=-1 && d < min_d) ){
            min_d=d;
            min_i=i;
        }
    }

    return(&point_set[min_i]);
}

```

5. Problem name: Rectangle_Area_Ratio

```

#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int id;
    int sx;
    int sy;
    int ex;
    int ey;
    int area;
} RECTANGLE;

int main()
{
    RECTANGLE *rect;
    int i,total_area,N;
    double area_ratio;

    fscanf(stdin,"%d",&N);
    rect = (RECTANGLE *)malloc(N*sizeof(RECTANGLE));
    if(rect==NULL)
        return -1;

    total_area=0;
    for(i=0;i<N;i++){
        fscanf(stdin,"%d %d %d %d %d",&(rect[i].id),&(rect[i].sx),
            &(rect[i].sy),&(rect[i].ex),&(rect[i].ey));
        rect[i].area=abs(rect[i].ex-rect[i].sx)*abs(rect[i].ey-rect[i].sy);
        total_area=total_area+rect[i].area;
    }

    for(i=0;i<N;i++)
    {
        area_ratio=(double)rect[i].area/total_area;
        printf("%d %d %d %d %d %d %.2f\n",rect[i].id,rect[i].sx,
            rect[i].sy,
            rect[i].ex,
            rect[i].ey,
            rect[i].area,area_ratio);
    }

    free(rect);
    return 0;
}

```

6. Problem name: Count_Article

Note: เฉลยของโจทย์ข้อนี้มีการใช้ standard library function อยู่หลายฟังก์ชัน ซึ่งไม่ได้สอนในชั้นเรียน แต่โปรแกรมนี้สามารถเขียนโดยไม่ใช่ standard library function ก็ได้

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_STR_LEN 500

int main()
{
    char instr[MAX_STR_LEN+1];
    char word[MAX_STR_LEN+1];
    int count,i,j,k,len,first,last,state;
    int A_count,An_count,The_count;

    gets(instr);

    A_count=0;
    An_count=0;
    The_count=0;
    state=0;
    i=0;
    while(instr[i]!=0)
    {
        if(state==0 )
        {
            if(instr[i]!=' ')
            {
                first=i;
                state=1;
            }
        }
        else if(state==1)
        {
            if(instr[i]==' ')
            {
                last=i-1;
                state=0;

                for(k=0,j=first;j<=last;j++)
                {
                    word[k]=tolower(instr[j]);
                    k++;
                }
                word[k]=0;

                if(strcmp(word,"a")==0)
                    A_count++;
                else if(strcmp(word,"an")==0)
                    An_count++;
                else if(strcmp(word,"the")==0)
                    The_count++;
            }
        }
        i++;
    }
}
```

```

if(state==1)
{
    last=i-1;
    state=0;

    for(k=0,j=first;j<=last;j++)
    {
        word[k]=tolower(instr[j]);
        k++;
    }
    word[k]=0;

    if(strcmp(word,"a")==0)
        A_count++;
    else if(strcmp(word,"an")==0)
        An_count++;
    else if(strcmp(word,"the")==0)
        The_count++;

}

printf("%d\n",A_count);
printf("%d\n",An_count);
printf("%d\n",The_count);

return 0;
}

```