

CP 111 Computer Programming

2/2554 Section B01/B02

Lab 8 Solution

1. Problem name: Extract_Date

```
#include <stdio.h>
#include <math.h>

void Extract_Date(int date, int *day, int *month, int *year);

int main()
{
    int i_date, dd, mm, yyyy;
    scanf("%d", &i_date);
    Extract_Date (i_date, &dd, &mm, &yyyy);
    printf("%d\n", dd);
    printf("%d\n", mm);
    printf("%d\n", yyyy);
    return 0;
}

void Extract_Date(int date, int *day, int *month, int *year)
{
    *month=date/1000000;
    date=date-(*month*1000000);
    *day=date/10000;
    *year=date-(*day*10000);
}
```

2. Problem name: Quadratic_equation

```
#include <stdio.h>
#include <math.h>

void Quad_Eq(double a, double b, double c, double *sol1, double *sol2);

int main()
{
    double a,b,c,x1,x2;

    scanf("%lf %lf %lf",&a,&b,&c);
    Quad_Eq(a,b,c, &x1, &x2);
    printf("%.2f\n",x1);
    printf("%.2f\n",x2);
    return 0;
}

void Quad_Eq(double a, double b, double c, double *sol1, double *sol2)
{
    double sqrt_term;
    double tmp;

    sqrt_term=sqrt( (b*b) - (4*a*c));
    *sol1= (-b - sqrt_term)/(2*a);
    *sol2= (-b + sqrt_term)/(2*a);
    if(*sol1 > *sol2)
    {
        tmp=*sol1;
        *sol1=*sol2;
        *sol2=tmp;
    }
}
```

3. Problem name: Min_Max_Avg_Std_Func

```
#include <stdio.h>
#include <math.h>
void Min_Max_Avg_Std_Func_Pointer(int *,int,int *,int *,double *,double *);

int main()
{
    int N,x,i,min,max,ptr;
    double avg,std;
    int data[100];

    N=0;
    do{
        scanf("%d",&x);
        if(x>=0){
            data[N]=x;
            N=N+1;
        }
    }
    while(x>=0);
    ptr=data;
    Min_Max_Avg_Std_Func_Pointer(data, N, &min,&max,&avg,&std);
    printf("%d\n",min);    printf("%d\n",max);
    printf("%.2f\n",avg);    printf("%.2f\n",std);
    return 0;
}

void Min_Max_Avg_Std_Func_Pointer(int *pData,int N, int *pMin,
                                   int *pMax, double *pAvg, double *pStd)
{
    int i,sum;
    double sum_sq;

    sum=0;
    for(i=0;i<N;i++){
        if(i==0){
            *pMin=pData[i];
            *pMax=pData[i];
        }
        else{
            if(pData[i] < *pMin)
                *pMin=pData[i];
            if(pData[i] > *pMax)
                *pMax=pData[i];
        }
        sum=sum+pData[i];
    }
    *pAvg=(double)sum/N;
    sum_sq=0.0;
    for(i=0;i<N;i++)
        sum_sq=sum_sq+( (pData[i]-*pAvg)* (pData[i]-*pAvg) );
    *pStd=sqrt(sum_sq/N);
}
```

4. Problem name: Concat_String

```
#include <stdio.h>
#include <string.h>
#define MAX_STR_LEN 100

void concat_string(char *outstr, char *instr);

int main()
{
    char str1[MAX_STR_LEN+1], str2[MAX_STR_LEN+1], str3[MAX_STR_LEN+1];
    char ostr[MAX_STR_LEN+MAX_STR_LEN+MAX_STR_LEN+1];

    gets(str1);
    gets(str2);
    gets(str3);

    ostr[0]=0;
    concat_string(ostr, str1);
    concat_string(ostr, str2);
    concat_string(ostr, str3);
    printf("%s\n", ostr);

    ostr[0]=0;
    concat_string(ostr, str3);
    concat_string(ostr, str2);
    concat_string(ostr, str1);
    printf("%s\n", ostr);

    return 0;
}

void concat_string(char *outstr, char *instr)
{
    int i, k, last;

    i=0;
    while(outstr[i]!=0)
        i++;

    k=0;
    while(instr[k]!=0)
    {
        ostr[i]=instr[k];
        i++;
        k++;
    }
    ostr[i]=0;
}
```

5. Problem name: Pick_two_cards

```
#include <stdio.h>

int main()
{
    int i,j,X[10];
    int X1,X2,minsum,sum;

    for(i=0;i<10;i++){
        scanf("%d",&X[i]);
    }

    minsum=10000;
    for(i=0;i<10;i++)
    {
        for(j=0;j<10;j++)
        {
            if(i!=j)
            {
                sum=X[i]+X[j];
                if(sum<minsum && sum>=10)
                {
                    minsum=sum;
                    X1=X[i];
                    X2=X[j];
                }
            }
        }
    }

    if(X1<X2)
        printf("%d %d %d\n",X1,X2,minsum);
    else
        printf("%d %d %d\n",X2,X1,minsum);

    return 0;
}
```