

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

Programming Homework 09 solution

1. Problem name: Recur_GCD_Func

```
#include <stdio.h>

int Recur_GCD(int m, int n);

int main()
{
    int a,b,gcd;

    scanf("%d %d",&a,&b);

    gcd=Recur_GCD(a,b);

    printf("%d\n",gcd);

    return 0;
}

int Recur_GCD(int m, int n)
{
    if(m==0)
    {
        return(n);
    }
    else if(n==0)
    {
        return(m);
    }
    else
    {
        return(Recur_GCD(n,m%n));
    }
}
```

2. Problem name: Recur_Maximum

```
#include <stdio.h>
int Recur_Maximum(int *A, int n);
int main()
{
    int data[100], i, n, max;

    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &data[i]);
    max=Recur_Maximum(data, n);
    printf("%d\n", max);
    return 0;
}
int Recur_Maximum(int *A, int n)
{
    int x;
    if(n==1)
        return A[0];
    else
    {
        x=Recur_Maximum(A, n-1);
        if(x>A[n-1])
            return x;
        else
            return A[n-1];
    }
}
```

3. Problem name: Recur_Power_Func

```
#include <stdio.h>
double Recur_Power(double x, int p);
int main()
{
    int pp;
    double xx, value;

    scanf("%lf %d", &xx, &pp);
    value=Recur_Power(xx, pp);
    printf("%.2f\n", value);
    return 0;
}
double Recur_Power(double x, int p)
{
    if(p==0)
        return(1);
    else
        return(x*Recur_Power(x, p-1));
}
```

4. Problem name: Recur_ReverseStr

```
#include <stdio.h>
void Recur_ReverseStr(char *str,int len);
int main()
{
    int len,i;
    char str[1001];
        gets(str);
        i=0;
        while(str[i]!=0)
            i=i+1;
        len=i;
        Recur_ReverseStr(str,len);
        printf("%s\n",str);
        return 0;
}
void Recur_ReverseStr(char *str,int len)
{
    char tmp;
        if(len<=1)
            return;
        else
        {
            tmp=str[0];
            str[0]=str[len-1];
            str[len-1]=tmp;
            Recur_ReverseStr(str+1,len-2);
        }
}
```

5. Problem name: Recur_Even_Factorial

```
#include <stdio.h>
int Recur_eFact(int n);
int main()
{
    int n,val;
        scanf("%d",&n);
        val= Recur_eFact(n);
        printf("%d\n",val);
        return 0;
}

int Recur_eFact(int n)
{
    if(n<2)
        return 1;
    if(n%2==1)
        return(Recur_eFact(n-1));
    else
    {
        if(n==2)
            return 2;
        else
            return(n*Recur_eFact(n-2));
    }
}
```

6. Problem name: Recur_Palindrome

```
#include <stdio.h>

int Recur_IsPalindrome(char *str,int len);

int main()
{
    int i,k,len,retval;
    char str[501];

    for(k=0;k<5;k++)
    {
        gets(str);

        // find string length
        i=0;
        while(str[i]!=0)
            i++;
        len=i;

        retval=Recur_IsPalindrome(str,len);

        if(retval==1)
            printf("YES\n");
        else
            printf("NO\n");
    }
    return 0;
}

int Recur_IsPalindrome(char *str,int len)
{
    if(len<=1)
        return 1;

    if(str[0]==' ' && str[len-1]==' ')
        return(Recur_IsPalindrome(str+1,len-2));
    else if(str[0]==' ' && str[len-1]!=' ')
        return(Recur_IsPalindrome(str+1,len-1));
    else if(str[0]!=' ' && str[len-1]==' ')
        return(Recur_IsPalindrome(str,len-1));
    else
    {
        if(str[0]==str[len-1])
            return(Recur_IsPalindrome(str+1,len-2));
        else
            return 0;
    }
}
```

7. Problem name: Iter_Palindrome

```
#include <stdio.h>
int Iter_IsPalindrome(char *str);
int main()
{
    int i,k,len,retval;
    char str[501];

    for(k=0;k<5;k++)
    {
        gets(str);
        retval=Iter_IsPalindrome(str);
        if(retval==1)
            printf("YES\n");
        else
            printf("NO\n");
    }
    return 0;
}

int Iter_IsPalindrome(char *str)
{
    int i,j,k;

    j=0; // index of the leftmost character
    i=0;
    while(str[i]!=0)
        i++;
    k=i-1; // index of the rightmost character

    while(1)
    {
        if(j>=k)
            return 1;
        if(str[j]==' ' && str[k]==' ')
        {
            j=j+1;
            k=k-1;
        }
        else if(str[j]==' ' && str[k]!=' ')
            j=j+1;
        else if(str[j]!=' ' && str[k]==' ')
            k=k-1;
        else
        {
            if(str[j]!=str[k])
                return 0;
            j=j+1;
            k=k-1;
        }
    }
}
```