

第8章 指针

(本章习题要求用指针方法处理)

8-1

```
#include <stdio.h>
void swap(int *p1,int *p2)
{int p;
    p=*p1; *p1=*p2; *p2=p;
}
int main()
{ int n1,n2,n3;
    int *p1,*p2,*p3;
    printf("input three integer n1,n2,n3:");
    scanf("%d,%d,%d",&n1,&n2,&n3);
    p1=&n1;
    p2=&n2;
    p3=&n3;
    if(n1>n2) swap(p1,p2);
    if(n1>n3) swap(p1,p3);
    if(n2>n3) swap(p2,p3);
    printf("Now,the order is:%d,%d,%d\n",n1,n2,n3);
    return 0;
}
```

8-3

```
#include <stdio.h>
void input(int *number)
{int i;
    printf("input 10 numbers:");
    for (i=0;i<10;i++)
        scanf("%d",&number[i]);
}

void max_min_value(int *number)
{ int *max,*min,*p,temp;
    max=min=number;
    for (p=number+1;p<number+10;p++)
        if (*p>*max) max=p;
        else if (*p<*min) min=p;
    temp=number[0];number[0]=*min;*min=temp;
    if(max==number) max=min;
    temp=number[9];number[9]=*max;*max=temp;
```

不使用指针时:

```
void max_min_value(int number[])
{int max,maxI,min,minI,temp,i;
    max=min=number[0];
    maxI=minI=0;
    for(i=1;i<10;i++)
        if (max<number[i]) {max=number[i]; maxI=i;}
        else if (min>number[i]){min=number[i]; minI=i;}
    temp=number[0];number[0]=number[minI];number[minI]=temp;
    if(maxI==0){ temp=minI;minI=maxI;maxI=temp;}
    temp=number[9];number[9]=number[maxI];number[maxI]=temp;
}
```

```

    }

void output(int *number)
{
    int *p;
    printf("Now,they are:   ");
    for (p=number;p<number+10;p++)
        printf("%d ",*p);
    printf("\n");
}

int main()
{
    int number[10];
    input(number);
    max_min_value(number);
    output(number);
    return 0;
}

```

8-6

```

#include <stdio.h>
int length(char *p)
{
    int n;
    n=0;
    while (*p!='\0')
    {
        n++;
        p++;
    }
    return(n);
}

int main()
{
    int len;
    char str[20];
    printf("input string:  ");
    scanf("%s",str);
    len=length(str);
    printf("The length of string is %d.\n",len);
    return 0;
}

```

8-7

```

void copystr(char *p1,char *p2,int m)
{
    int n=0;
    while(n<m-1)
    {
        n++;
    }
}

```

```

        p1++;
    }
    while(*p1!='\0')
    { *p2=*p1;
      p1++;
      p2++;
    }
    *p2='\0';
}
main()
{ int m;
  char str1[20],str2[20];
  printf("input string:");
  gets(str1);
  printf("Which character that begin to copy?");
  scanf("%d",&m);
  if(strlen(str1)<m)printf("input error!");
  else {
      copystr(str1,str2,m);
      printf("result:%s",str2);
  }
}

```

8-12

```

#define N 10
void sort (char *p[])
{ int i,j;
  char *temp;
  for(i=0;i<N-1;i++)
      for(j=i+1;j<N;j++)
          if(strcmp(p[i],p[j])>0)
          { temp=p[i];
            p[i]=p[j];
            p[j]=temp;
          }
}
main()
{ int i;
  char *p[N],str[N][20];
  for(i=0;i<N;i++)p[i]=str[i];
  printf("Input strings;");
  for(i=0;i<N;i++)scanf("%s",p[i]);
}

```

```

    sort(p);
    printf("Now, the sequence is:\n");
    for(i=0;i<N;i++)printf("%s\n",p[i]);
}

```

8-14

方法一、

```

#include "stdio.h"
#include <malloc.h>
void inv(int a[], int n)
{
    int i,t;
    for (i = 0; i<n / 2; i++)
    {
        t = a[i]; a[i] = a[n - i - 1]; a[n - i - 1] = t;
    }
}

int main()
{
    int n, *a, i, t;
    printf("Input n:");
    scanf("%d", &n);
    a = (int *)malloc(n * sizeof(int));
    printf("\nInput a[i]:\n");
    for (i = 0; i<n; i++)
    {
        scanf("%d", &a[i]);
        printf("%d ", a[i]);
    }
    inv(a, n);
    printf("\n逆序后\n");
    for (i = 0; i<n; i++)
        printf("%d ", a[i]);
    free(a);
    return 0;
}

```

方法二(不用函数)、

```

#include "stdio.h"
#include <malloc.h>
main()
{ int n,*a,i,t;
    printf("Input n:");
    scanf("%d",&n);

```

```

a=(int *)malloc(n*sizeof(int));
printf("\nInput a[i]:\n");
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
    printf("%d ",a[i]);
}
printf("\n");
for(i=0;i<n/2;i++)
{
    t=a[i];a[i]=a[n-i-1];a[n-i-1]=t;
}
for(i=0;i<n;i++)
{
    printf("%d ",a[i]);
}
free(a);
}

```