MỘT SỐ ĐÁP ÁN BÀI TẬP MẪU CHƯƠNG 2

```
//Bai01aChuong2.cpp
//Viet chuong trinh nhan vao mot day cac so nguyen nhap tu ban phim.
//Luu tru danh sach theo thu tu nhap vao.
//In danh sach ra mang hinh.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
```

```
        printf("Vi tri khong hop le");
    else{
         Position q;

         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    Position p=1;
    printf("\nEnter a list of interger numbers. -1 to stop.\n");
    do{
        printf("x=");scanf("%d",&x);
        if(x!=-1)
           Insert_List(x,EndList(L),L);
    }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
    printf("\n");
}

main(){
    List L;
    ElementType x;
    Position p;
    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L);
    getch();
}
```

```
//Bai01bChuong2.cpp
//Viet chuong trinh nhan vao mot day cac so nguyen nhap tu ban phim.
//Luu tru danh sach theo thu tu nguoc voi thu tu nhap vao.
//In danh sach ra mang hinh.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;
         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    Position p=1;
    printf("\nEnter a list of interger numbers. -1 to stop.\n");
    do{
        printf("x=");scanf("%d",&x);
        if(x!=-1)
            Insert_List(x,First(L),L);
    }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
    printf("\n");
}

main(){
    List L;
    ElementType x;
    Position p;
    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L);
    getch();
}
```

```
//Bai03aChuong2.cpp
//Viet chuong trinh con sap xep mot danh sach tang dan.
//Dung danh sach dac.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;
         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Ham swap hoan doi noi dung 2 phan tu o vi tri p va q*/
void swap(Position p, Position q, List &L){
     ElementType temp;
     temp=L.Elements[p-1];
     L.Elements[p-1]=L.Elements[q-1];
     L.Elements[q-1]=temp;
}

/*Ham Sort sap xep danh sach theo thu tu tang dan*/
void Sort(List &L){
     Position p=First(L);
     //vi tri phan tu dau tien trong danh sach
     while(p!=EndList(L)){
             Position q=Next(p,L);
             //vi tri phan tu dung sau phan tu p
             while(q!=EndList(L)){
                if(Retrieve(p,L)>Retrieve(q,L))
                    swap(p,q,L);
                q=Next(q,L);
             }
             p=Next(p,L);
     }
}

/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    Position p=1;
    printf("\nEnter a list of interger numbers. -1 to stop.\n");
    do{
        printf("x=");scanf("%d",&x);
        if(x!=-1)
            Insert_List(x,First(L),L);
    }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
}

main(){
    List L;
    ElementType x;
    Position p;
    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L); printf("\n");
    Sort(L);
    printf("Danh sach sau khi sap xep:");
    Print_List(L); printf("\n");
    getch();
}
```

//Bai03bChuong2.cpp
//Viet chuong trinh con sap xep mot danh sach tang dan.
//Dung danh sach lien ket.

```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
        return P->Next->Element;
}
```

```
/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Sap xep tang dan*/
void Sort(List &L){
   Position P,Q;
   ElementType temp;
   P=L->Next;
   while(P!=NULL){
      Q=P->Next;
      while(Q!=NULL){
         if(P->Element > Q->Element){
             temp=P->Element;
             P->Element=Q->Element;
             Q->Element=temp;
         }
         Q=Q->Next;
       }
       P=P->Next;
   }
}


/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");
   MakeNull_List(L);
   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
           Insert_List(x,EndList(L),L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}
```

```
main(){
    List L;
    ElementType x;
    Position p;
    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L); printf("\n");
    Sort(L);
    printf("Danh sach sau khi sap xep:");
    Print_List(L); printf("\n");
    getch();
}
```

```
//Bai04Chuong2.cpp
//Viet chuong trinh con them mot phan tu vao danh sach da co thu tu
//sao cho ta van co danh sach co thu tu.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}
```

```
/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Ham them mot phan tu vao danh sach co thu tu tang,
sao cho ta van co danh sach co thu tu tang*/
Position Insert_In_Ordered_List(ElementType X, List L){
   Position P;

   //Chon vi tri thich hop P de xen vao
   P=L;
   while((P->Next!=NULL)&&(P->Next->Element<X))
      P=P->Next;

   //Xen X vao vi tri P cua danh sach L
   Insert_List(X,P,L);

}
/*Sap xep danh sach tang dan*/
void Sort(List &L){
   Position P,Q;
   ElementType temp;
   P=L->Next;
   while(P!=NULL){
      Q=P->Next;
      while(Q!=NULL){
         if(P->Element > Q->Element){
             temp=P->Element;
             P->Element=Q->Element;
             Q->Element=temp;
         }
         Q=Q->Next;
       }
       P=P->Next;
   }
}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   int i,n;
   printf("So phan tu cua danh sach la: "); scanf("%d",&n);
   for(i=1;i<=n;i++){
       printf("x=");scanf("%d",&x);
       Insert_List(x,EndList(L),L);
   }
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   Sort(L);
   printf("\nDanh sach L sau khi sap xep tang dan la:"); Print_List(L);

   printf("\nNhap vao mot phan tu de them vao danh sach, x=");
   scanf("%d",&x);
   Insert_In_Ordered_List(x,L);
   printf("\nDanh sach L sau khi them %5d la:",x); Print_List(L);

   getch();
}
```

```
//Bai05Chuong2
//Viet chuong trinh tim kiem va
//xoa mot phan tu trong danh sach co thu tu.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>

typedef int ElementType;

typedef struct Node{
   ElementType Element;
   struct Node *Next;
};

typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Xoa 1 phan tu khoi danh sach*/
void Delete_List(Position P, List &L){
   Position T;
   if(P->Next!=NULL){
      T=P->Next;
      P->Next=T->Next;
      free(T);
   }
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}
```

```
/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Xoa bo mot phan tu trong danh sach co thu tu*/
void Delete_In_Ordered_List(ElementType X, List &L){
   Position P;

   //Tim kiem tu dau danh sach cho den khi tim gap phan tu dau tien >= X
   P=L;
   while((P->Next!=NULL)&&(P->Next->Element<X))
      P=P->Next;

   //Xoa phan tu tim gap
   if((P->Next!=NULL)&&(P->Next->Element==X))
         Delete_List(P,L);
}

/*Sap xep tang dan*/
void Sort(List &L){
   Position P,Q;
   ElementType temp;
   P=L->Next;
   while(P!=NULL){
      Q=P->Next;
      while(Q!=NULL){
         if(P->Element > Q->Element){
             temp=P->Element;
             P->Element=Q->Element;
             Q->Element=temp;
         }
         Q=Q->Next;
       }
       P=P->Next;
   }
}
```

```
/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   int i,n;
   printf("So phan tu cua danh sach la: "); scanf("%d",&n);
   for(i=1;i<=n;i++){
       printf("x=");scanf("%d",&x);
       Insert_List(x,EndList(L),L);
   }
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   Sort(L);
   printf("\nDanh sach L sau khi sap xep tang dan la:"); Print_List(L);

   printf("\nNhap vao mot phan tu de xoa, x="); scanf("%d",&x);
   Delete_In_Ordered_List(x,L);
   printf("\nDanh sach L sau khi xoa %5d la:",x); Print_List(L);

   getch();
}
```

```
//Bai06aChuong2.cpp
//Viet chuong trinh con nhan tu ban phim mot danh sach cac so nguyen.
//Luu tru vao danh sach theo thu tu khong giam.
//Dung danh sach dac.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;
```

```
/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;

         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Dinh vi va chen mot phan tu trong danh sach co thu tu tang*/
void Insert_In_Ordered_List(ElementType x, List &L){
    Position p;
    int Found=0;

    //Tim vi tri thich hop p de xen
    p=First(L);
    while((p!=EndList(L))&&(Retrieve(p,L)<x))
        p=Next(p,L);

   //Xen x vao vi tri p
   Insert_List(x,p,L);

}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;

   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");
   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
          Insert_In_Ordered_List(x,L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
    printf("\n");
}

main(){
   ElementType x;
   int n;
   List L;
   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   printf("\nDanh sach L vua nhap duoc luu theo thu tu khong giam la:");
   Print_List(L);
   getch();
}
```

```
//Bai06bChuong2.cpp
//Viet chuong trinh con nhan tu ban phim mot danh sach cac so nguyen.
//Luu tru vao danh sach theo thu tu khong giam.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
```

```
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Dinh vi va chen mot phan tu vao trong danh sach co thu tu tang*/
void Insert_In_Ordered_List(ElementType X, List L){
   Position P;

   //Tim vi tri thich hop P de xen
   P=L;
   while((P->Next!=NULL)&&(P->Next->Element<X))
      P=P->Next;

   //Xen X vao vi tri P
   Insert_List(X,P,L);

}
```

```
/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;

   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");
   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
          Insert_In_Ordered_List(x,L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L;
   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   printf("\nDanh sach L vua nhap duoc luu theo thu tu khong giam la:");
   Print_List(L);
   getch();
}
```

```
//Bai07aChuong2.cpp
//Viet chuong trinh con loai bo cac phan tu trung nhau
//(chi giu lai mot phan tu) trong mot danh sach co thu tu.
//Dung danh sach dac.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}
```

```
/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;

         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Xoa 1 phan tu ra khoi danh sach*/
void Delete_List(Position p, List &L){
    if((p<1)||(p>L.Last))
        printf("Vi tri khong hop le");
    else if(Empty_List(L))
            printf("Danh sach rong!");
    else{   Position q;
            for(q=p-1;q<L.Last-1;q++)
            L.Elements[q]=L.Elements[q+1];
            L.Last--;
     }
}
```

```
/*Ham swap hoan doi noi dung 2 phan tu o vi tri p va q*/
void swap(Position p, Position q, List &L){
     ElementType temp;
     temp=L.Elements[p-1];
     L.Elements[p-1]=L.Elements[q-1];
     L.Elements[q-1]=temp;
}

/*Ham Sort sap xep danh sach theo thu tu tang dan*/
void Sort(List &L){
     Position p=First(L);
     //vi tri phan tu dau tien trong danh sach
     while(p!=EndList(L)){
             Position q=Next(p,L);
             //vi tri phan tu dung sau phan tu p
             while(q!=EndList(L)){
                if(Retrieve(p,L)>Retrieve(q,L))
                    swap(p,q,L);
                q=Next(q,L);
             }
             p=Next(p,L);
     }
}

/*Ham loai bo cac phan tu trung nhau (chi giu lai mot phan tu)
trong mot danh sach co thu tu*/
void Remove_Duplicates(List &L){
     Position p=First(L);
     //vi tri phan tu dau tien trong danh sach
     while(p!=EndList(L)){
             Position q=Next(p,L);
             //vi tri phan tu dung sau phan tu p
             while(q!=EndList(L))
                if(Retrieve(p,L)==Retrieve(q,L))
                    Delete_List(q,L);
                else q=Next(q,L);

             p=Next(p,L);
     }
}

/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    Position p=1;
    printf("\nEnter a list of interger numbers. -1 to stop.\n");
    do{
        printf("x=");scanf("%d",&x);
        if(x!=-1)
           Insert_List(x,EndList(L),L);
    }while(x!=-1);
}
```

```
/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
}

main(){
    List L;
    ElementType x;
    Position p;

    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L); printf("\n");

    Sort(L);
    printf("Danh sach sau khi sap xep:");
    Print_List(L); printf("\n");

    Remove_Duplicates(L);
    printf("Danh sach sau khi loai bo cac phan tu trung nhau:");
    Print_List(L);
    getch();
}
```

```
//Bai07bChuong2.cpp
//Viet chuong trinh con loai bo cac phan tu trung nhau
//(chi giu lai mot phan tu) trong mot danh sach co thu tu.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}
/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}
```

```
/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Xoa 1 phan tu khoi danh sach*/
void Delete_List(Position P, List &L){
   Position T;
   if(P->Next!=NULL){
      T=P->Next;
      P->Next=T->Next;
      free(T);
   }
}

/*Dinh vi mot phan tu trong danh sach*/
Position Locate(ElementType X, List L){
   Position P;
   int Found=0;
   P=L;
   while((P->Next!=NULL)&&(Found==0))
       if(P->Next->Element==X)Found=1;
       else P=P->Next;
   return P;
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Sap xep tang dan*/
void Sort(List &L){
   Position P,Q;
   ElementType temp;
   P=L->Next;
   while(P!=NULL){
      Q=P->Next;
      while(Q!=NULL){
         if(P->Element > Q->Element){
             temp=P->Element;
             P->Element=Q->Element;
             Q->Element=temp;
         }
         Q=Q->Next;
       }
       P=P->Next;
   }
}

/*Ham loai bo cac phan tu trung nhau (chi giu lai mot phan tu)
trong mot danh sach co thu tu*/
void Remove_Duplicates(List &L){
     Position p=L;
     //vi tri phan tu dau tien trong danh sach
     while(p->Next!=NULL){
             Position q=p->Next;
             //vi tri phan tu dung sau phan tu p
             while(q->Next!=NULL)
                if(Retrieve(p,L)==Retrieve(q,L))
                    Delete_List(q,L);
                else q=q->Next;
             p=p->Next;
     }
}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");
   MakeNull_List(L);
   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
          Insert_List(x,EndList(L),L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}
```

```
main(){
    List L;
    ElementType x;
    Position p;

    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L); printf("\n");

    Sort(L);
    printf("Danh sach sau khi sap xep:");
    Print_List(L); printf("\n");

    Remove_Duplicates(L);
    printf("Danh sach sau khi loai bo cac phan tu trung nhau:");
    Print_List(L);
    getch();
}
```

```
//Bai08aChuong2.cpp
//Viet chuong trinh con nhan tu ban phim mot day so nguyen,luu tru no
//trong danh sach co thu tu tang, khong co hai phan tu trung nhau.
//Dung danh sach dac.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;

         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Dinh vi 1 phan tu trong danh sach*/
Position Locate(ElementType x, List L){
    Position p;
    int Found=0;

    p=First(L);
    while((p!=EndList(L))&&(Found==0))
         if(Retrieve(p,L)==x)Found=1;
         else p=Next(p,L);
    return p;
}

/*Dinh vi va chen mot phan tu trong danh sach co thu tu tang*/
void Insert_In_Ordered_List_Not_Duplicates(ElementType x, List &L){
    Position p;
    int Found=0;

    //Neu chua tim thay phan tu x trong danh sach
    //thi moi them no vao danh sanh
    if(Locate(x,L)==EndList(L)){
        //Tim vi tri thich hop p de xen x vao
        p=First(L);
        while((p!=EndList(L))&&(Retrieve(p,L)<x))
            p=Next(p,L);
```

```
       //Xen x vao vi tri p
       Insert_List(x,p,L);
     }
}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");

   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
          Insert_In_Ordered_List_Not_Duplicates(x,L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
    printf("\n");
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);

   printf("\nDanh sach duoc sap va loai bo hai phan tu trung nhau la:\n");
   Print_List(L);
   getch();
}
```

```
//Bai08bChuong2.cpp
//Viet chuong trinh con nhan tu ban phim mot day so nguyen,luu tru no
//trong mot danh sach co thu tu tang khong co hai phan tu trung nhau.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
```

```
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Dinh vi mot phan tu trong danh sach*/
Position Locate(ElementType X, List L){
   Position P;
   int Found=0;
   P=L;
   while((P->Next!=NULL)&&(Found==0))
       if(P->Next->Element==X)Found=1;
       else P=P->Next;
   return P;
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Dinh vi va chen mot phan tu vao trong danh sach co thu tu tang*/
Position Insert_In_Ordered_List(ElementType X, List L){
   Position P;

   //Tim vi tri thich hop P de xen
   P=L;
   while((P->Next!=NULL)&&(P->Next->Element<X))
      P=P->Next;

   //Xen X vao vi tri P
   Insert_List(X,P,L);

}

/*Dinh vi va chen mot phan tu trong danh sach co thu tu tang*/
void Insert_In_Ordered_List_Not_Duplicates(ElementType x, List &L){
    Position p;
    int Found=0;

    //Neu chua tim thay phan tu x trong danh sach
    //thi moi them no vao danh sanh
    if(Locate(x,L)==EndList(L)){
        //Tim vi tri thich hop p de xen x vao
        p=First(L);
        while((p!=EndList(L))&&(Retrieve(p,L)<x))
            p=Next(p,L);

       //Xen x vao vi tri p
       Insert_List(x,p,L);
     }
}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;

   printf("Nhap vao 1 danh sach cac so nguyen, -1 de dung\n");
   do{
       printf("x=");scanf("%d",&x);
       if(x!=-1)
          Insert_In_Ordered_List_Not_Duplicates(x,L);
   }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
       printf("%5d",P->Element);
       P=P->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);

   printf("\nDanh sach duoc sap va loai bo hai phan tu trung nhau la:\n");
   Print_List(L);
   getch();
}
```

```
//Bai09Chuong2.cpp
//Viet chuong trinh tron 2 danh sach cac so nguyen theo thu tu tang.
//de duoc mot danh sach cung co thu tu tang.
//Dung danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   ElementType Element;
   struct Node *Next;
};
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}
```

```
/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Sap xep danh sach L tang dan*/
void Sort(List &L){
   Position P,Q;
   ElementType temp;
   P=L->Next;
   while(P!=NULL){
      Q=P->Next;
      while(Q!=NULL){
         if(P->Element>Q->Element){
             temp=P->Element;
             P->Element=Q->Element;
             Q->Element=temp;
         }
         Q=Q->Next;
       }
       P=P->Next;
   }
}

/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   int i,n;
   printf("So phan tu cua danh sach la: "); scanf("%d",&n);
   for(i=1;i<=n;i++){
       printf("x=");scanf("%d",&x);
       Insert_List(x,EndList(L),L);
   }
}
```

```
/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}

/*Tron 2 danh sach dang co thu tu tang, L1, L2,
de duoc danh sach co thu tu tang L3*/
void Merge_Two_Ordered_Lists(List L1, List L2, List &L3){
   Position P1,P2,P3;

   MakeNull_List(L3);
   P1=L1->Next; P2=L2->Next; P3=L3;

   //Chon phan tu nho hon trong L1 va L2 de xen vao L3
   while(P1!=NULL && P2!=NULL){
       if(P1->Element<P2->Element){
           Insert_List(P1->Element,P3,L3);
           P1=P1->Next;
           }
       else {
           Insert_List(P2->Element,P3,L3);
           P2=P2->Next;
           }
       P3=P3->Next;
   }

   //Xen het phan con lai cua L1 vao L3
   while(P1!=NULL){
       Insert_List(P1->Element,P3,L3);
       P1=P1->Next;
       P3=P3->Next;
   }

   //Xen het phan con lai cua L2 vao L3
   while(P2!=NULL){
       Insert_List(P2->Element,P3,L3);
       P2=P2->Next;
       P3=P3->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L1, L2, L3;

   MakeNull_List(L1);
   printf("\nNhap vao danh sach L1:\n"); Read_List(L1);
   Sort(L1);
   printf("\nDanh sach L1 sau khi sap xep tang dan la:"); Print_List(L1);

   MakeNull_List(L2);
   printf("\nNhap vao danh sach L2:\n"); Read_List(L2);
   Sort(L2);
   printf("\nDanh sach L2 sau khi sap xep tang dan la:"); Print_List(L2);

   Merge_Two_Ordered_Lists(L1,L2,L3);
   printf("\nDanh sach L3 thu duoc sau khi tron L1 va L2:");
   Print_List(L3);

   getch();
}
```

```
//Bai10aChuong2.cpp
//Viet chuong trinh xoa khoi danh sach luu tru cac so nguyen le.
//Cai dat bang danh sach dac.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}
```

```
/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;

         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Xoa 1 phan tu ra khoi danh sach*/
void Delete_List(Position p, List &L){
    if((p<1)||(p>L.Last))
        printf("Vi tri khong hop le");
    else if(Empty_List(L))
            printf("Danh sach rong!");
    else{
            Position q;
            for(q=p-1;q<L.Last-1;q++)
            L.Elements[q]=L.Elements[q+1];
            L.Last--;
     }
}

/*Xoa bo cac so nguyen le*/
void Delete_Odd_Numbers(List &L){
   Position P;

   //Tim kiem va xoa cac phan tu chua so nguyen le
   P=First(L);
   while(P!=EndList(L)){
      if(Retrieve(P,L)%2==1)
          Delete_List(P,L);
      else
          P=Next(P,L);
    }
}
/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    Position p=1;
    printf("\nEnter a list of interger numbers. -1 to stop.\n");
    do{
        printf("x=");scanf("%d",&x);
        if(x!=-1)
           Insert_List(x,EndList(L),L);
    }while(x!=-1);
}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
    printf("\n");
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   printf("\nDanh sach L vua nhap la:"); Print_List(L);

   Delete_Odd_Numbers(L);
   printf("\nDanh sach L sau khi xoa cac so le:"); Print_List(L);

   getch();
}
```

```
//Bai10bChuong2.cpp
//Viet chuong trinh xoa khoi danh sach luu tru cac so nguyen le.
//Cai dat bang danh sach lien ket.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>

typedef int ElementType;

typedef struct Node{
   ElementType Element;
   struct Node *Next;
};

typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(ElementType X, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->Element=X;
   T->Next=P->Next;
   P->Next=T;
}

/*Xoa 1 phan tu khoi danh sach*/
void Delete_List(Position P, List &L){
   Position T;
   if(P->Next!=NULL){
      T=P->Next;
      P->Next=T->Next;
      free(T);
   }
}

/*Tra ve gia tri cua phan tu o vi tri P*/
ElementType Retrieve(Position P, List L){
   if(P->Next!=NULL)
       return P->Next->Element;
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu ke tiep*/
Position Next(Position P, List L){
   return P->Next;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Xoa bo cac so nguyen le*/
void Delete_Odd_Numbers(List &L){
   Position P;
   //Tim kiem va xoa cac phan tu chua so nguyen le
   P=First(L);
   while(P->Next!=NULL){
      if(Retrieve(P,L)%2==1)
          Delete_List(P,L);
      else
          P=P->Next;
    }
}


/*Nhap danh sach*/
void Read_List(List &L){
   ElementType x;
   int i,n;
   printf("So phan tu cua danh sach la: "); scanf("%d",&n);
   for(i=1;i<=n;i++){
       printf("x=");scanf("%d",&x);
       Insert_List(x,EndList(L),L);
   }
}

/*In danh sach ra man hinh*/
void Print_List(List L){
   Position P;
   P=L->Next;
   while(P!=NULL){
      printf("%5d",P->Element);
      P=P->Next;
   }
}

main(){
   ElementType x;
   int n;
   List L;

   MakeNull_List(L);
   printf("\nNhap vao danh sach L:\n"); Read_List(L);
   printf("\nDanh sach L vua nhap la:"); Print_List(L);

   Delete_Odd_Numbers(L);
   printf("\nDanh sach L sau khi xoa cac so le:"); Print_List(L);

   getch();
}
```

```
//Bai11Chuong2.cpp
//Viet chuong trinh nhan vao mot day cac so nguyen nhap tu ban phim.
//Viet chuong trinh con tach mot danh sach chua cac so nguyen thanh 2
//danh sach: Mot danh sach gom cac chan. Mot danh sach gom cac so le.

#include <conio.h>
#include <stdio.h>
#define MaxLength 100
typedef int ElementType;
typedef int Position;

typedef struct{
   ElementType Elements[MaxLength];
   Position Last;
}List;

/*Khoi tao danh sach rong*/
void MakeNull_List(List &L){
   L.Last=0;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return L.Last==0;
}

/*Vi tri phan tu dau tien*/
Position First(List L){
    return 1;
}

/*Gia tri phan tu o vi tri p*/
ElementType Retrieve(Position p, List L){
    return L.Elements[p-1];
}

/*Vi tri sau vi tri phan tu cuoi cung*/
Position EndList(List L){
    return L.Last+1;
}

/*Vi tri sau vi tri p*/
Position Next(Position p, List L){
    return p+1;
}

/*Xen 1 phan tu vao danh sach*/
void Insert_List(ElementType x, Position p, List &L){
    if(L.Last==MaxLength)
        printf("Danh sach day");
    else if((p<1)||(p>L.Last+1))
        printf("Vi tri khong hop le");
    else{
         Position q;
         for(q=(L.Last-1)+1;q>p-1; q--)
             L.Elements[q]=L.Elements[q-1];

         L.Elements[p-1]=x;
         L.Last++;
         }
}

/*Doc vao 1 danh sach*/
void Read_List(List &L){
    ElementType x;
    int i, n;
    printf("\nNhap vao so phan tu trong danh sach:"); scanf("%d",&n);
    for(i=1;i<=n;i++){
        printf("x=");scanf("%d",&x);
        Insert_List(x,EndList(L),L);
    }
}

/*Ham tach danh sach L thanh 2 danh sach: L1 chua so chan va L2 chua so
le*/
void SeparateEvenOddNumbers(List L, List &L1, List &L2){
    ElementType x;
    Position p=First(L);
    //vi tri phan tu dau tien trong danh sach
    MakeNull_List(L1); MakeNull_List(L2);
    while(p!=EndList(L)){
        x=Retrieve(p,L);
        if(x%2==0) Insert_List(x,EndList(L1),L1);
        else Insert_List(x,EndList(L2),L2);
        p=Next(p,L);
    }

}

/*In danh sach ra man hinh*/
void Print_List(List L){
    Position p;
    for(p=1;p!=EndList(L);p++){
        printf("%5d",Retrieve(p,L));
    }
}

main(){
    List L, L1, L2;
    ElementType x;
    Position p;
    MakeNull_List(L);
    Read_List(L);
    printf("Danh sach vua nhap:");
    Print_List(L); printf("\n");
    SeparateEvenOddNumbers(L,L1,L2);
    printf("Danh sach chua cac so chan la:");
    Print_List(L1); printf("\n");
    printf("Danh sach chua cac so le la:");
    Print_List(L2); printf("\n");
    getch();
}
```

```
//Bai13Chuong2.cpp
//Da thuc P(x)=an*x^n+an_1*x^n-1 + ... + a1*x + a0
//Duoc luu tru trong may duoi dang mot danh sach lien ket
//ma moi phan tu co 3 truong: he so, so mu va next.
//Viet chuong trinh thuc hien viec luu tru nay.
//Viet chuong trinh con de cong hai da thuc.
//Viet chuong trinh con tinh dao ham cua da thuc.

#include <stdio.h>
#include <conio.h>
#include <malloc.h>
typedef int ElementType;
typedef struct Node{
   float heso;
   int somu;
   struct Node *Next;
};
typedef struct Node* Position;
typedef Position List;

/*Tao danh sach rong*/
void MakeNull_List(List &Header){
   Header=(Node*)malloc(sizeof(Node));
   Header->Next=NULL;
}

/*Kiem tra danh sach rong*/
int Empty_List(List L){
   return (L->Next==NULL);
}

/*Xen mot phan tu vao danh sach*/
void Insert_List(float hso, int mu, Position P, List &L){
   Position T;
   T=(Node*)malloc(sizeof(Node));
   T->heso=hso;
   T->somu=mu;
   T->Next=P->Next;
   P->Next=T;
}

/*Xoa 1 phan tu khoi danh sach*/
void Delete_List(Position P, List &L){
   Position T;
   if(P->Next!=NULL){
      T=P->Next;
      P->Next=T->Next;
      free(T);
   }
}

/*Xac dinh vi tri phan tu dau*/
Position First(List L){
   return L;
}

/*Xac dinh vi tri phan tu sau phan tu cuoi*/
Position EndList(List L){
   Position P;
   P=First(L);
   while(P->Next!=NULL)P=P->Next;
   return P;
}

/*Sap xep da thuc giam dan theo so mu*/

void Sort(List &L){
   Position P,Q;
   float temp_heso;
   int temp_somu;

   for(P=L->Next;P!=NULL;P=P->Next){
      for(Q=P->Next;Q!=NULL;Q=Q->Next){
         if(P->somu < Q->somu){
             temp_heso=P->heso; temp_somu=P->somu;
             P->heso=Q->heso; P->somu=Q->somu;
             Q->heso=temp_heso; Q->somu=temp_somu;
         }//if
       }//for
   }//for
}

/*Nhap danh sach*/
void Read_List(List &L){
   float a;
   int n;
   char traloi;

   MakeNull_List(L);
   printf("Nhap vao cac he so va so mu cua da thuc, nhan n/N de dung.\n");
   while(1){
       printf("\nNhap he so: "); scanf("%f",&a);
       printf("\nNhap so mu: "); scanf("%d",&n);
       Insert_List(a,n,EndList(L),L);
       printf("Ban co muon tiep tuc nhap da thuc khong (y/n)? ");
       fflush(stdin); traloi=getchar();
       if(traloi!='y' && traloi!='Y')
          break;
   }//while

}
```

```
/*In da thuc ra man hinh*/
void Print_List(List L){
   Position P;

   for(P=L->Next;P!=NULL;P=P->Next){
      if(P->heso<0)
          printf(" - %.2fx^%d", (-1)*(P->heso), P->somu);
      else if(P->heso>0)
          printf(" + %.2fx^%d", P->heso, P->somu);
      else
          continue;
   }//for
}

/*Ham tinh tong cua hai da thuc*/
void Sum_Two_Polies(List L1, List L2, List &L3){
     Position P1,P2;

     P1=L1->Next; P2=L2->Next;
     MakeNull_List(L3);

     while(P1!=NULL && P2!=NULL)
        if(P1->somu > P2->somu){
             Insert_List(P1->heso,P1->somu,EndList(L3),L3);
             P1=P1->Next;
        }else if(P1->somu < P2->somu){
             Insert_List(P2->heso,P2->somu,EndList(L3),L3);
             P2=P2->Next;
        }else{
            Insert_List(P1->heso+P2->heso,P1->somu,EndList(L3),L3);
            P1=P1->Next; P2=P2->Next;
        }

     while(P1!=NULL){
        Insert_List(P1->heso,P1->somu,EndList(L3),L3);
        P1=P1->Next;
     }

     while(P2!=NULL){
        Insert_List(P2->heso,P2->somu,EndList(L3),L3);
        P2=P2->Next;
     }
}

/*Ham tinh dao ham cua da thuc*/
void Derivation(List L, List &L_kq){
   Position P;

   MakeNull_List(L_kq);
   for(P=L->Next;P!=NULL;P=P->Next){
       float heso_kq=P->heso*P->somu;
       int somu_kq=P->somu-1;
       if(heso_kq!=0)
```

```
           Insert_List(heso_kq,somu_kq,EndList(L_kq),L_kq);
   }//for
}

main(){
   List L1, L2;
   int chon;

   printf("Nhap vao da thuc thu nhat:\n");
   Read_List(L1); Sort(L1);

   printf("\nNhap vao da thuc thu hai:\n");
   Read_List(L2); Sort(L2);

   while(1){
      printf("\nBan hay chon:\n");
      printf("\n1. Hien thi hai da thuc.");
      printf("\n2. Cong hai da thuc.");
      printf("\n3. Tinh dao ham hai da thuc.");
      printf("\n4. Thoat.");
      printf("\nChon: "); scanf("%d",&chon);

      switch(chon){
          case 1:
                printf("\nDa thuc thu nhat la: ");Print_List(L1);
                printf("\nDa thuc thu hai la: ");Print_List(L2);
                break;
          case 2:
                List L3;
                Sum_Two_Polies(L1,L2,L3);
                printf("\nDa thuc tong la: ");Print_List(L3);
                break;
          case 3:
               List L1_kq, L2_kq;
               Derivation(L1,L1_kq); Derivation(L2,L2_kq);
               printf("\nDao ham cua da thuc thu nhat la: ");
                Print_List(L1_kq);
               printf("\nDao ham cua da thuc thu hai la: ");
               Print_List(L2_kq);
          default:
                break;
      }//switch

      if(chon==4)
         break;
   }//while

   getch();
}
```