ch4

【1】課本code 4.16

【2】課本code 4.17

【3】The delete() function with 2 pointers

```
void delete(listPointer *first, listPointer trail)
{/* deletefrom the list, trail is the preceding node
and *first is the front of the list */
if (trail)
   trail->link = (*first)->link;
else
  *first = (*first)->link;
free(x);
}
```

【4】search() function

```
Call:
if (search(ptr,key))
     printf("The key is in the list\n");
else
  printf("The key is not in the list\n");
list_pointer search(list_pointer ptr, int key)
{/* determine if key is in the list */
  list_pointer temp;
  for (temp = ptr; temp; temp = temp->link)
        if (temp->item.key == key) return temp;
  return NULL;
}
```

【5】length() function

```
Call:
   printf("The list contains %4d elements:\n\n",length(ptr));

int length(list_pointer ptr)
{/* find the length of the list */
  list_pointer temp;
  int size;
  size = 0;
  for (temp=ptr; temp; temp = temp->link)
     size++;
  return size;
}
```

【6】

```
Call:   a =  ReadPoly();

PolyPointer ReadPoly()
{/*read the polynomial into a chain */
  PolyPointer front, rear,temp;
  float coefficient;
  int exponent;
  front=rear=NULL;
  printf("Enter an exponent Less than 0 to quit: \n");
  printf("Coefficient, Exponent: ");
  scanf("%f,%d",&coefficient,&exponent);
  while (exponent >= 0) {
     temp = (PolyPointer)malloc(sizeof(struct PolyNode));
     temp->coef =  coefficient;
     temp->expon = exponent;
     temp->link = NULL;
     if (!front) front = temp;
     else rear->link = temp;
     rear = temp;
     printf("Coefficient, Exponent: ");
     scanf("%f,%d",&coefficient,&exponent);
     }
     return front;
}
```

【7】

```
Call:   result = evalPoly(x0,a)

float evalPoly(float x0, PolyPointer ptr)
{/*evaluate the polynomial at point x */
  PolyPointer temp;
  float result = 0;
  for (temp = ptr; temp; temp= temp->link)
    result = result + temp->coef * pow(x0,temp->expon);
  return result;
}
```

【8】

```
Call:    a =  ReadPoly();

PolyPointer ReadPoly()
{/*read in the polynomial */
  PolyPointer node,c;
  float coefficient;
  int exponent;

  node = GetNode();
  node->coef = -1.0;
  node->expon = -1;
  node->link = node;
  printf("Enter an exponent < 0 to quit: ");
  printf("\nCoefficient, Exponent: ");
  scanf("%f,%d",&coefficient,&exponent);
  while (exponent >= 0) {
     c = GetNode();
     c->coef =  coefficient;
     c->expon = exponent;
     c->link = node->link;
     node->link = c;
     printf("Coefficient, Exponent: ");
     scanf("%f,%d",&coefficient,&exponent);
  }
  return node;
}
```

【9】找不到