

4.9

a.

Procedure AddPolynomial(A,B, S)

n = A[1];

m = B[1];

if n >= m then

Begin

S[1] = n;

For i = 2 To (n - m + 1) Do

S[i] = A[i];

For i = 2 To m + 2 Do

S[n - m + i] = A[n - m + i] + B[i];

End;

Else

Begin

S[1] = m;

For i = 1 To (m - n + 1) Do

S[i] = B[i];

For i = 2 To n + 2 Do

Begin

S[m - n + i] = B[m - n + i] + A[i];

End;

End;

END.

b.

{Tìm hệ số của  $x^k$  trong array A}

Function GetCoefficientForTerm(A, k);

For i = 1 To A[1] Do

Begin

If A[2\*i] = k Then

Begin

c = A[2\*i+1];

Return c;

End;

End;

Return 0 ;

Procedure AddPolynomials(A, B, S);

Size = 1;

i = MAX(A[2], B[2]); { tìm hệ số lớn nhất của 2 đa thức }

While i >= 0 Do

Begin

SUM = GetCoefficientForTerm(A, i) + GetCoefficientForTerm(B, i);

If SUM > 0 Then

Begin

```
S[2*Size] = i;  
S[2*Size + 1] = SUM;  
Size = Size + 1;  
End;  
i = i - 1;  
End;  
S[1] = Size;  
END.
```

4.10

1. Tính số các nút trong danh sách

Function DEM (L)

```
p= L; dem = 0;  
while p < > NULL do begin  
    dem= dem + 1;  
    p= NEXT(p);  
end.  
return dem.
```

2. Tìm nút thứ k trong danh sách

Function TIMK (L,k)

```
p= L; i = 1;  
while (i < k ) and (p < > NULL) do begin  
    i= i + 1;  
    p= NEXT(p) ;  
end.  
return p.
```

3. Bỏ sung một nút vào sau nút thứ k

Procedure BOSUNGSAUK(L, X,k)

```
call New(p) ; INFO(p) = X;  
if L = NULL then begin NEXT(p) = NULL; L= p; return ; end.  
q = TIMK(L,k) ;  
if q < > NULL then begin  
    NEXT(p) = NEXT(q) ;  
    NEXT(q) = p;  
end;
```

4. Loại bỏ nút đứng trước nút thứ k

Procedure LOAIBO(L, k)

```
p= L; q= r= NULL; i = 1;  
while (i<k) and p < > NULL then begin  
    i= i + 1;  
    r= q; q= p; p= NEXT(p) ;  
end;  
if r < > NULL then NEXT(r) = p;
```

## Cấu trúc dữ liệu và Giải thuật – Bài tập chương 4

```
else if  $q \neq \text{NULL}$  then  $L = p$ ;  
else write('Danh sách có 1 nút' );  
Call Dispose (q);  
return.
```

5. Chèn danh sách trở bởi P vào sau nút trở bởi M trong danh sách

6. Tách danh sách thành 2 danh sách con, nút đầu danh sách con là nút trở bởi M

Procedure TACH(L,M)

```
p = L;  
while NEXT(p)  $\neq$  M do  $p = \text{NEXT}(p)$  ;  
NEXT(p) = NULL;
```

7. Đảo ngược danh sách đã cho (danh sách mới trở bởi L')

Procedure DAO(L);

```
    r = L;  
    q = null;  
    While r  $\neq$  NULL Do  
        Begin  
            p = q;  
            q = r ; r = NEXT(r); NEXT(q) := p;  
        End;  
    l' = q;  
END.
```