MỘT SỐ ĐÁP ÁN BÀI TẬP MẪU CHƯƠNG 3 (PHẦN 1)

```
//Bai1Chuong3.cpp
//Viet chuong trinh xay dung 1 cay tong quat de luu tru cac ky tu.
//- Nhap vao so nut cua cay.
//- Ung voi moi nut thi phai nhap nhan va cha cua no.
//- Hien thu danh sach duyet cay theo cac thu tu tien tu, trung tu va hau tu

#include <conio.h>
#include <stdio.h>
#define MAXLENGTH 1000 //chieu dai toi da cua mang
#define NIL -1

typedef char DataType;
typedef int Node;

typedef struct{
    /*Luu tru nhan (du lieu) cua nut trong cay*/
    DataType Data[MAXLENGTH];
    /*Luu tru cha cua cac nut trong cay theo nguyen tac: Cha cua nut i se luu o
vi tri i trong mang */
    Node Parent[MAXLENGTH];
   /*So nut thuc su trong cay*/
    int MaxNode;
} Tree;

Tree T;

/*Khoi tao cay rong*/
void MakeNull_Tree (Tree *T){
   (*T).MaxNode=0;
}

/*Kiem tra cay rong*/
int EmptyTree(Tree T)
{
    return T.MaxNode == 0;
}

/*Xac dinh nut cha cua nut tren cay*/
Node Parent(Node n,Tree T){
  if(EmptyTree(T)||(n>T.MaxNode-1))
        return NIL;
   else return T.Parent[n];
}

/*Xac dinh nhan cua nut tren cay*/
DataType Label_Node(Node n,Tree T){
  if(!EmptyTree(T)&&(n<=T.MaxNode-1))
   return T.Data[n];
}

/*Ham xac dinh nut goc trong cay*/
Node Root(Tree T){
  if (!EmptyTree(T)) return 0;
  else return NIL;
}
```

```
/*Ham xac dinh con trai nhat cua mot nut*/
Node LeftMostChild(Node n,Tree T){
   Node i; int found;
   if (n<0) return NIL;
    i=n+1; /*Vi tri nut dau tien hy vong la con cua nut n*/
   found=0;
   while ((i<=T.MaxNode-1) && !found)
      if (T.Parent[ i] ==n) found=1;
      /* tim thay con trai nhat cua nut n */
      else i=i+1;
   if (found) return i;
   else return NIL;
}

/*Ham xac dinh anh em ruot phai cua mot nut*/
Node RightSibling(Node n,Tree T){
   Node i,parent;  int found;
   if (n<0) return NIL;
   parent=T.Parent[ n] ;
   i=n+1;
   found=0;
   while ((i<=T.MaxNode-1) && !found)
      if (T.Parent[ i] ==parent) found=1;
      else i=i+1;
   if (found) return i;
   else return NIL;
}

/*Thu tuc duyet tien tien tu*/
void PreOrder(Node n,Tree T){
   Node i;
   printf("%c ",Label_Node(n,T));
   i=LeftMostChild(n,T);
   while (i!=NIL){
       PreOrder(i,T);
       i=RightSibling(i,T);
     }
}

/*Thu tuc duyet trung tu*/
void InOrder(Node n,Tree T) {
   Node i;
   i=LeftMostChild(n,T);
   if (i!=NIL) InOrder(i,T);
   printf("%c ",Label_Node(n,T));
   i=RightSibling(i,T);
   while (i!=NIL){
      InOrder(i,T);
      i=RightSibling(i,T);
    }
}

/*Thu tuc duyet hau tu*/
void PostOrder(Node n,Tree T){
   Node i;
   i=LeftMostChild(n,T);
   while (i!=NIL){
      PostOrder(i,T);
      i=RightSibling(i,T);
    }
   printf("%c ",Label_Node(n,T));
}
```

```
/*Thu nhap cay tu ban phim*/
void ReadTree(Tree *T){
  int i;
  MakeNull_Tree(&*T);
  do{  printf("Nhap so nut ");
       scanf("%d",&(*T).MaxNode);
  }while (((*T).MaxNode<1) || ((*T).MaxNode>MAXLENGTH));

  printf("Nhap nhan cua nut goc "); fflush(stdin);
  scanf("%c",&(*T).Data[0]);
 (*T).Parent[0]=NIL; // nut goc khong co cha
  for (i=1;i<=(*T).MaxNode-1;i++){
     printf("Nhap cha cua nut %d ",i);
     scanf("%d",&(*T).Parent[i]);
     printf("Nhap nhan cua nut %d ",i);
     fflush(stdin);
     scanf("%c",&(*T).Data[i]);
  }
}

void main(){
  printf("Nhap du lieu cho cay tong quat\n");
  ReadTree(&T);
  printf("Danh sach duyet tien tu cua cay la\n");
  PreOrder(Root(T),T);
  printf("\nDanh sach duyet trung tu la\n");
  InOrder(Root(T),T);
  printf("\nDanh sach duyet hau tu cua cay la\n");
  PostOrder(Root(T),T);
  getch();
}
```

//Bai2Chuong3.c
//Viet chuong trinh xay dung 1 cay tong quat de luu tru cac ky tu.
//- Nhap vao so nut cua cay.
//- Ung voi moi nut thi phai nhap nhan va cha cua no.
//- Hien thu danh sach duyet cay theo muc

```
#include <stdio.h>
#include <conio.h>
#define Maxlength 100
#define NIL -1

/*****************************************/
/*Cai dat hang doi */
/*****************************************/
typedef int ElementType;
typedef struct{
      ElementType Elements[Maxlength];
      int Front, Rear;
}Queue;

/*Khoi tao 1 hang doi rong*/
void MakeNull_Queue(Queue *Q){
  Q->Front=-1;
  Q->Rear=-1;
}
/*Kiem tra xem 1 hang doi co rong khong*/
int Empty_Queue(Queue Q){
  return Q.Front==-1;
}
```

```
/*Kiem tra xem 1 hang doi co day khong*/
int Full_Queue(Queue Q){
  return (Q.Rear-Q.Front+1) % Maxlength == 0;
}

/*Xoa bo 1 phan tu khoi hang doi*/
void DeQueue(Queue *Q){
  if (!Empty_Queue(*Q)){
    //Neu hang chi chua mot phan tu thi khoi tao lai hang rong luon
    if (Q->Front==Q->Rear) MakeNull_Queue(Q);
    else Q->Front=(Q->Front+1) % Maxlength;
    //tang Front lời 1 don vi
  }
  else printf("Loi: Hang rong!");
}

/*Them 1 phan tu vao hang*/
void EnQueue(ElementType X,Queue *Q){
  if (!Full_Queue(*Q)){
    if (Empty_Queue(*Q)) Q->Front=0;
    Q->Rear=(Q->Rear+1) % Maxlength;
    Q->Elements[Q->Rear]=X;
  }
  else printf("Loi: Hang day!");
}

/*Tra ve phan tu dau hang*/
ElementType Front(Queue Q){
       if(Empty_Queue(Q))
             printf("Hang rong!");
       else return Q.Elements[Q.Front];
}


/*****************************************/
/*Cai dat cay tong quat de luu cac ky tu*/
/*****************************************/
typedef char DataType;
typedef int Node;

typedef struct{
    /*Luu tru nhan (du lieu) cua nut trong cay*/
    DataType Data[Maxlength];
    /*Luu tru cha cua cac nut trong cay theo nguyen tac: Cha cua nut i se
      luu o vi tri i trong mang */
    Node Parent[Maxlength];
   /*So nut thuc su trong cay*/
    int MaxNode;
} Tree;

Tree T;

/*Khoi tao cay rong*/
void MakeNull_Tree (Tree *T){
   (*T).MaxNode=0;
}

/*Kiem tra xem 1 cay co rong khong*/
int EmptyTree(Tree T){
    return T.MaxNode == 0;
}
```

```
/*Xac dinh nut cha cua 1 nut tren cay*/
Node Parent(Node n,Tree T){
  if(EmptyTree(T)||(n>T.MaxNode-1))
      return NIL;
   else return T.Parent[n];
}

/*Xac dinh nhan cua 1 nut tren cay*/
DataType Label_Node(Node n,Tree T){
  if(!EmptyTree(T)&&(n<=T.MaxNode-1))
      return T.Data[n];
}

/*Ham xac dinh nut goc trong cay*/
Node Root(Tree T){
  if (!EmptyTree(T)) return 0;
  else return NIL;
}

/*Ham xac dinh con trai nhat cua mot nut*/
Node LeftMostChild(Node n,Tree T){
   Node i; int found;
   if (n<0) return NIL;
   i=n+1; /*Vi tri nut dau tien hy vong la con cua nut n*/
   found=0;
   while ((i<=T.MaxNode-1) && !found)
   if (T.Parent[i]==n) found=1;
    /* tim thay con trai nhat cua nut n */
   else i=i+1;
   if (found) return i;
   else return NIL;
}

/*Ham xac dinh anh em ruot phai cua mot nut*/
Node RightSibling(Node n,Tree T){
   Node i,parent;  int found;
   if (n<0) return NIL;
   parent=T.Parent[n];
   i=n+1;
   found=0;
   while ((i<=T.MaxNode-1) && !found)
      if (T.Parent[i]==parent) found=1;
      else i=i+1;
   if (found) return i;
   else return NIL;
}

/*Ham nhap vao 1 cay tu ban phim*/
void ReadTree(Tree *T){
  int i;
  MakeNull_Tree(&*T);
  do{ printf("Nhap so nut ");
       scanf("%d",&(*T).MaxNode);
  }while (((*T).MaxNode<1) || ((*T).MaxNode>Maxlength));

  printf("Nhap nhan cua nut goc "); fflush(stdin);
  scanf("%c",&(*T).Data[0]);
  (*T).Parent[0]=NIL; // nut goc khong co cha

  for (i=1;i<=(*T).MaxNode-1;i++){
     printf("Nhap cha cua nut %d ",i);
       scanf("%d",&(*T).Parent[i]);
```

```
      printf("Nhap nhan cua nut %d ",i);
      fflush(stdin);
      scanf("%c",&(*T).Data[i]);
   }
}

/*Ham duyet cay theo muc*/
void levelorder(Node n, Tree T){
      Queue Q; Node i;

      /*Tao mot hang doi rong*/
      MakeNull_Queue(&Q);
      i=n;
      if(i!=NIL)EnQueue(i,&Q);

      while(!Empty_Queue(Q)) {
              /*in ra nut i*/
              i=Front(Q);
              printf("%5c",Label_Node(i,T));
              DeQueue(&Q);

             /*dua cac con cua nut i (tu trai sang phai) vao hang doi*/
             i=LeftMostChild(i,T);
             while(i!=NIL){
                    EnQueue(i,&Q);
                    i=Rightsibling(i,T);
             } //while
       }//while
}//void

void main(){
         printf("Nhap du lieu cho cay tong quat, chua cac ky tu\n");
         ReadTree(&T);
         printf("\nDuyet cay theo muc:");
         levelorder(Root(T),T);
         getch();
}
```

//Bai04aChuong3.c
//Viet chuong trinh de tinh gia tri cho bieu thuc tien to.
//Vi du: Dau vao (input): * + 6 4 5
//thi dau ra (output) se la 50 vi bieu thuc tren la dang tien to cua 6+4) * 5.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

#define Maxlength 100

/*Cai dat ngan xep*/
typedef int ElementType;
typedef struct{
    ElementType Elements[Maxlength];
    int Top_idx;
}Stack;

/*Khoi tao 1 ngan xep rong*/
void MakeNull_Stack(Stack *S){
  S->Top_idx=Maxlength;
}
```

```
 /*Kiem tra xem 1 ngan xep co rong khong*/
int Empty_Stack(Stack S){
  return (S.Top_idx==Maxlength);
}

/*Kiem tra xem 1 ngan xep co day khong*/
int Full_Stack(Stack S){
  return (S.Top_idx==0);
}

/*Ham tra ve noi dung cua phan tu tren dinh*/
ElementType Top(Stack S){
  if(!Empty_Stack(S))
    return S.Elements[ S.Top_idx] ;
  else printf("Error! Stack is empty");
}

/*Xoa 1 phan tu khoi ngan xep*/
void Pop(Stack *S){
  if(!Empty_Stack(*S))
    S->Top_idx=S->Top_idx+1;
  else printf("Error! Stack is empty");
}

/*Day 1 phan tu len ngan xep*/
void Push(ElementType x, Stack *S){
  if(Full_Stack(*S))
    printf("Error!Stack is full");
  else{
    S->Top_idx=S->Top_idx-1;
    S->Elements[ S->Top_idx] =x;
  }
}

/**Kiem tra mot so co co phai toan hang hay khong?*/
int laToanHang(char so[ ]){
    int dem=0;
    for(int i=0;i<strlen(so);i++)
        if(so[ i]>='0' && so[ i]<='9') dem++;

    if(dem==strlen(so)) return 1;
    else return 0;
}

/*Giai thuat Tinh tri cho bieu thuc tien to:
Lap lai viec doc tung token cua bieu thu phai sang trai cua bt tien to.
Voi moi token doc duoc:
- Neu no la toan hang thi push no vao stack
- Neu no la toan tu thi pop hai toan hang trong stack ra va tinh gia tri
  cua phep toan len hai toan hang nay. Push ket qua tro lai stack.
Cuoi cung, sau khi vong lap tren ket thuc, mot phan tu con lai tren dinh stack
dang luu gia tri cua bieu thuc tien to.
*/
```

```
float tinhBieuThucTienTo(char bttiento[]){
	/*tach cac token de luu vao 1 mang tokens*/
	const char phancach[2] = " ";
	char *token;
	char *tokens[100];

	int index = 0;
	token = strtok(bttiento, phancach);    //lay token dau tien
	while(token != NULL ) {
		tokens[index] = token;
		index++;
		//lay token ke tiep
		token = strtok(NULL, phancach);
	 }//while

	/* Kiem tra: in ra cac tu token theo thu tu dao nguoc de kiem tra xem
	for (int i = index-1; i>=0; i--)
		printf ("\n%s ", tokens[i]);
	 printf ("\n");
	*/

	//tao mot stack
	Stack S;
	MakeNull_Stack(&S);

	// duyet qua cac token phai sang trai cua bieu thu tien to
	for (int i = index-1; i>=0; i--) {
		if(laToanHang(tokens[i]))
			Push(atof(tokens[i]), &S);
		else {
			ElementType b = Top(S); Pop(&S);
			ElementType a = Top(S); Pop(&S);
			char pheptoan=tokens[i][0];
			switch (pheptoan) {
				case '+': b += a; break;
				case '-': b -= a; break;
				case '*': b *= a; break;
				case '/': b /= a; break;
			 }//switch
			 Push(b, &S);
		  }//else
	}//for
	return Top(S);
}

void main(){

   ElementType x;
   char bttiento[50];

   printf("Nhap vao 1 bieu thu tien to, trong do toan hang va toan tu cach nhau
1 khoang trang.\n");
   printf("(vi du: * + 6 4 5 cho (6+4)*5, gia tri la 50:"); gets(bttiento);
   printf("Gia tri cua bieu thu hau to tren la: %.2f",
tinhBieuThucTienTo(bttiento));

   getch();
}
```

```
//Bai04bChuong3.cpp
//Viet chuong trinh de tinh gia tri cho bieu thuc hau to.
//Vi du: Dau vao (input): 6 4 5 + *
//thi dau ra (output) se la 54 vi bieu thuc tren la dang hau to cua 6*(4+5).

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define Maxlength 100

/*Cai dat ngan xep*/
typedef int ElementType;
typedef struct{
  ElementType Elements[ Maxlength] ;
  int Top_idx;
} Stack;

/*Khoi tao 1 ngan xep rong*/
void MakeNull_Stack(Stack *S){
  S->Top_idx=Maxlength;
}

/*Kiem tra xem 1 ngan xep co rong khong*/
int Empty_Stack(Stack S){
  return (S.Top_idx==Maxlength);
}

/*Kiem tra xem 1 ngan xep co day khong*/
int Full_Stack(Stack S){
  return (S.Top_idx==0);
}

/*Ham tra ve noi dung cua phan tu tren dinh*/
ElementType Top(Stack S){
  if(!Empty_Stack(S))
    return S.Elements[ S.Top_idx] ;
  else printf("Error! Stack is empty");
}

/*Xoa 1 phan tu khoi ngan xep*/
void Pop(Stack *S){
  if(!Empty_Stack(*S))
    S->Top_idx=S->Top_idx+1;
  else printf("Error! Stack is empty");
}

/*Day 1 phan tu len ngan xep*/
void Push(ElementType x, Stack *S){
  if(Full_Stack(*S))
    printf("Error!Stack is full");
  else{
    S->Top_idx=S->Top_idx-1;
    S->Elements[ S->Top_idx] =x;
  }
}
```

```
/**Kiem tra mot so co co phai toan hang hay khong?*/
int laToanHang(char so[]){
    int dem=0;
      int i;
    for(i=0;i<strlen(so);i++)
        if(so[i]>='0' && so[i]<='9') dem++;

    if(dem==strlen(so)) return 1;
    else return 0;
}

/*Giai thuat:
Lap lai viec doc tung token cua bieu thu hau tu tu trai sang phai.
Voi moi token doc duoc:
- Neu no la toan hang thi push no vao stack
- Neu no la toan tu thi pop hai toan hang trong stack ra va tinh gia tri
  cua phep toan len hai toan hang nay. Push ket qua tro lai stack.
Cuoi cung, sau khi vong lap tren ket thuc, mot phan tu con lai tren dinh stack
dang luu gia tri cua bieu thuc hau to.
*/

 float tinhBieuThucHauTo(char bthauto[]){

    const char phancach[2] = " ";
    char *token;

    /*tao stak rong*/
    Stack S;
    MakeNull_Stack(&S);

    /* duyet qua cac token tu trai sang phai cua bt hau to*/

    /* lay token dau tien */
    token = strtok(bthauto, phancach);

    while( token != NULL ) {
           if(laToanHang(token))
                 Push(atof(token), &S);
           else {
                    ElementType a = Top(S); Pop(&S);
                    ElementType b = Top(S); Pop(&S);

                    char pheptoan=token[0];
                    switch (pheptoan){
                              case '+': b += a; break;
                              case '-': b -= a; break;
                              case '*': b *= a; break;
                              case '/': b /= a; break;
                     }//switch

                    Push(b, &S);
             }//else

            //lay token ke tiep
            token = strtok(NULL, phancach);
     }//while

    return Top(S);
}
```

```
 main(){

	ElementType x;
	char bthauto[ 50] ;

	printf("Nhap vao 1 bieu thu hau to, trong do toan hang va toan tu cach
nhau 1 khoang trang.\n");
	printf("(vi du: 6 4 5 + * cho 6*(4+5), gia tri la 54):"); gets(bthauto);
	printf("Gia tri cua bieu thu hau to tren la: %.2f",
tinhBieuThucHauTo(bthauto));

	getch();
}
```