＜演習課題＞　リスト構造

【課題問題 1】

```
typedef struct {
   int   IntData;    /* データ：例えば会員番号のような正の数とする */
   int   NextIndex;  /* つながり情報：後続ノードの配列のインデックス */
} MyList;
```

上記の構造体の配列を用いた線形リストがある．各ノードは正の整数型データとつながり情報を持つ構造体である．ここで，繋がり情報とは後続ノードの格納されている配列インデックスの値である．つながり情報が -1 であるインデックス 2 は末尾ノードを示している.また，インデックス 5 以降はノードが無いことを意味する．

※先頭データのインデックスは 0 である

インデックス	0	1	2	3	4	5	6	7	8
データ	12	23	55	30	80	-1	-1	-1	-1
つながり情報	3	2	-1	4	1	-1	-1	-1	-1

次の(1)から(7)の解答を解答欄に記入せよ．

(1) つながり情報をもとにしてデータを並べよ．

(2) 42 のデータを 3 番目と 4 番目のノード間に挿入した．挿入後の構造体の内容を示せ．

(3) (2)に引き続き，2 番目のノードを削除した．削除後の構造体の内容を示せ．

(4) (3)に引き続き，25 のデータを末尾ノードの後ろに挿入した．挿入後の構造体の内容を示せ．

(5) (4)に引き続き，33 のデータを 4 番目と 5 番目のノード間に挿入した．挿入後の構造体の内容を示せ．

(6) (5)に引き続き，末尾ノードを削除した．削除後の構造体の内容を示せ．

(7) (6)の操作後のデータの並びを，繋がり情報をもとにして並べよ．

---------- 以下は解答欄 ----------

(1)

1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	7 番目	8 番目	9 番目

(2)

インデックス	0	1	2	3	4	5	6	7	8
データ									
つながり情報									

(3)

インデックス	0	1	2	3	4	5	6	7	8
データ									
つながり情報									

(4)

インデックス	0	1	2	3	4	5	6	7	8
データ									
つながり情報									

(5)

インデックス	0	1	2	3	4	5	6	7	8
データ									
つながり情報									

(6)

インデックス	0	1	2	3	4	5	6	7	8
データ									
つながり情報									

(7)

1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	7 番目	8 番目	9 番目

【課題問題 2】

配列のインデックスをつながり情報とした線形リストを表示するプログラムを以下の仕様を満たすように作成せよ.

1）各ノードのデータを正の整数として，リストを整数型配列とする．ノードのデータを，

2　4　6　8　10　12　14　16　18

として初期化子で設定する．配列のサイズは 9 より大きくし，データの入っていないノードすなわち配列要素には値-1 を代入しておく.

2）リストを表示する関数

```
void ShowList(int  data[])
```

を作り，リストの表示を行う．ただし，data[] はリストを表す配列である

※実行結果の例を右に示す.

```
256000@imac-000[41]: ./a.out
リスト：2  4  6  8  10  12  14  16  18
256000@imac-000[42]:
```

【課題問題 3】

課題問題 2 のプログラムを下の仕様を満たすように変更せよ.

1）**リストへのデータの挿入を行う関数**

```
int  InsertNode(int  no,  int  insdata,  int  data[])
```

を追加せよ. ただし, no は挿入するノードのリスト番号である. リスト番号は先頭ノードを 1 としたノードの順番を示す. insdata は挿入するデータであり, 正の整数である. data[]はリストを表す配列である. また, 挿入が成功したら戻り値 0, 配列が満杯等で挿入が失敗したら戻り値-1 を返し, main 関数で挿入の成功・失敗を表示すること.

2）**リストからデータを削除する関数**

```
int  DeleteNode(int  no,  int  data[])
```

を追加せよ. ただし, no は削除するノードのリスト番号である. また, 削除が成功したら戻り値 0, 削除対象がない等で失敗したら戻り値-1 を返し, main 関数で挿入の成功・失敗を表示すること.

3）データを初期化子でセットし, 最初にリストを表示してからデータの編集選択（挿入, 削除, 終了）を繰り返し行えるようにする. 挿入・削除後にはリストを表示する. 終了を選択するとプログラムを終了する. 実行例を以下に示す.

```
256000@imac-000[41]: ./a.out [Enter]
リスト：2  4  6  8  10  12  14  16  18
編集選択（挿入=i, 削除=d, 編集終了=q）： i [Enter]
挿入位置：4 [Enter]
挿入データ：7 [Enter]
リスト：2  4  6  7  8  10  12  14  16  18
編集選択（挿入=i, 削除=d, 編集終了=q）： d [Enter]
削除位置：7 [Enter]
リスト：2  4  6  7  8  10  14  16  18
編集選択（挿入=i, 削除=d, 編集終了=q）： q [Enter]
256000@imac-000[42]:
```

※なお, 編集選択の記号として, アルファベット i,d,q でなく整数を用いてもよい.

【課題問題 4】

課題問題 1 の構造体 MyList の配列を用いた線形リストを作り，以下の仕様を満たすように作成せよ．配列の大きさをマクロ定義で 20 として，リストの初期値を次のように設定する．6 つのノードを配列の先頭からインデックス 5 までの配列要素に格納している．インデックス 6 以降ではデータとつながり情報を-1 に初期化し，ノードが無いことを示す．つながり情報は後続ノードのインデックスを示す．インデックス 5 では，データは-1 でなく，つながり情報が-1 なので末尾ノードを示す．また，リストの先頭ノードはインデックス 0 に入れるものとする．

配列のインデックス	0	1	2	3	4	5	6	7	…
データ(IntData)	1	9	4	3	5	15	-1	-1	…
つながり情報(NextIndex)	3	2	5	4	1	-1	-1	-1	…

リストを表示する関数

```
void  ShowList(MyList  data[])
```

を作り，リストを表示せよ．
ここで data[] はリストを表す配列である．
※実行結果の例を右に示す．

```
256000@imac-000[41]: ./a.out
リスト：1  3  5  9  4  15
256000@imac-000[42]:
```

【課題問題 5】

課題問題 4 のプログラムを下の仕様を満たすように変更せよ．

1）**リストへのデータの挿入を行う関数**

```
int  InsertNode(int  no,  int  insdata,  MyList  data[])
```

を追加せよ．ただし，no は挿入するノードのリスト番号である．リスト番号は先頭ノードを 1 としたノードの順番を示す．insdata は挿入するデータであり，正の整数である．data[]はリストを表す配列である．また，挿入が成功したら戻り値 0，配列が満杯等で挿入が失敗したら戻り値-1 を返し，main 関数で挿入の成功・失敗を表示すること．

2）**リストからデータを削除する関数**

```
int  DeleteNode(int  no,  MyList  data[])
```

を追加せよ．ただし，no は削除するノードのリスト番号である．また，削除が成功したら戻り値 0，削除対象がない等で失敗したら戻り値-1 を返し，main 関数で挿入の成功・失敗を表示すること．

3）**リストからデータを取得する関数**

```
int getNode(int no, MyList data[])
```

を追加せよ．ただし，no は取得するノードのリスト番号である．リスト番号は先頭ノードを 1 としたノードの順番を示す．data[]はリストを表す配列である．取得が成功したらデータの値を返し，失敗した場合は戻り値-1 を返すこと．

4）実行直後に ShowList 関数を呼び出し最初にリストを表示してからデータの編集選択（挿入，削除，取得，終了）を繰り返し行えるようにする．挿入・削除後にも ShowList 関数を呼び出してリストを表示する．終了を選択するとプログラムを終了する．実行例を以下に示す．

```
256000@imac-000[41]: ./a.out [Enter]
リスト：1  3  5  9  4  15
編集選択（挿入=i，削除=d，取得=g，編集終了=q）： i [Enter]
挿入位置：4 [Enter]
挿入データ：7 [Enter]
リスト：1  3  5  7  9  4  15
編集選択（挿入=i，削除=d，取得=g，編集終了=q）： d [Enter]
削除位置：6 [Enter]
リスト：1  3  5  7  9  15
編集選択（挿入=i，削除=d，取得=g，編集終了=q）： g [Enter]
取得位置：2 [Enter]
3
編集選択（挿入=i，削除=d，取得=g，編集終了=q）： q [Enter]
256000@imac-000[42]:
```