Σ. Λίστα ταξινομημένη

NumberOfNodes

5

```
#define NilValue -1

typedef int ListElementType;

typedef int ListPointer;

typedef struct {
    ListElementType Data;
    ListPointer  Next;
} NodeType;
```

Node[ ]

FreePtr

0

	Data	Next
0	-1	1
1	-1	2
2	-1	3
3	-1	4
4	-1	-1

```
int main()
{
    ListPointer AList;
    NodeType Node[NumberOfNodes];
    ListPointer FreePtr,PredPtr;
    ListElementType AnItem;

    InitializeStoragePool(Node, &FreePtr);
    printAll(AList, FreePtr, Node);
    CreateList(&AList);           //DHMIOYRGIA LISTAS

    printf("DWSE ARI8MO GIA EISAGWGH STH LISTA: ");
    scanf("%d", &AnItem);
    PredPtr=NilValue;
    Insert(&AList, Node,&FreePtr, PredPtr, AnItem); //EI

    TraverseLinked(AList, Node);      //DIASXISH LISTAS
    printAll(AList, FreePtr, Node);

    return 0;
}
```

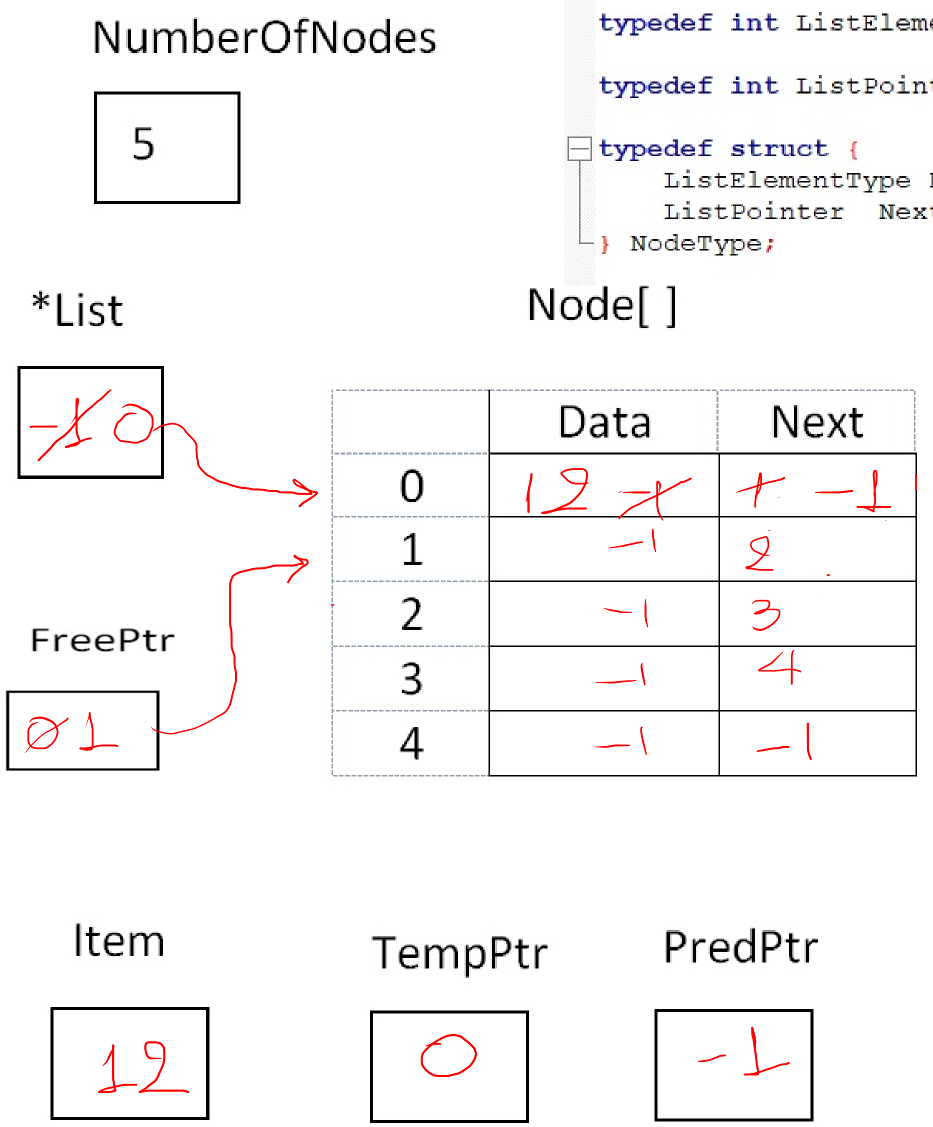
```
void InitializeStoragePool(NodeType Node[], ListPointer *FreePtr)
{
    int i;

    for (i=0; i<NumberOfNodes-1;i++)
    {
        Node[i].Next=i+1;
        Node[i].Data=-1;     /* δεν είναι αναγκαίο η απόδοση αρχικής
    }
    Node[NumberOfNodes-1].Next=NilValue;
    Node[NumberOfNodes-1].Data=-1;
    *FreePtr=0;
}
```

*List

-1

```
void CreateList(ListPointer *List)
{
  *List=NilValue;
}
```

Σ. Λίστα ταξινομημένη




```c
#define NilValue -1

typedef int ListElementType;

typedef int ListPointer;

typedef struct {
    ListElementType Data;
    ListPointer  Next;
} NodeType;
```

```c
int main()
{
    ListPointer AList;
    NodeType Node[NumberOfNodes];
    ListPointer FreePtr,PredPtr;
    ListElementType AnItem;

    InitializeStoragePool(Node, &FreePtr);
    printAll(AList, FreePtr, Node);
    CreateList(&AList);          //DHMIOYRGIA LISTAS

    printf("DWSE ARI8MO GIA EISAGWGH STH LISTA: ");
    scanf("%d", &AnItem);
    PredPtr=NilValue;
    Insert(&AList, Node,&FreePtr, PredPtr, AnItem); //EI

    TraverseLinked(AList, Node);      //DIASXISH LISTAS
    printAll(AList, FreePtr, Node);

    return 0;
}
```

```c
void Insert(ListPointer *List, NodeType Node[],ListPointer *FreePtr, ListPointer PredPtr, ListElementType Item)
{
  ListPointer TempPtr;
  GetNode(&TempPtr,FreePtr,Node);
  if (!FullLList(TempPtr)) {
    if (PredPtr==NilValue)
    {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =*List;
        *List =TempPtr;
    }
    else
      {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =Node[PredPtr].Next;
        Node[PredPtr].Next =TempPtr;
      }
  }
  else
    printf("Full List ...\n");
}
```

Σ. Λίστα ταξινομημένη

NumberOfNodes

5

```
#define NilValue -1

typedef int ListElementType;

typedef int ListPointer;

typedef struct {
    ListElementType Data;
    ListPointer  Next;
} NodeType;
```

Node[ ]

*List

1

FreePtr

1

	Data	Next
0	12	-1
1	9	~~20~~
2	-1	3
3	-1	4
4	-1	-1

Item

9

TempPtr

1

PredPtr

-1

```
int main()
{
    ListPointer AList;
    NodeType Node[NumberOfNodes];
    ListPointer FreePtr,PredPtr;
    ListElementType AnItem;

    InitializeStoragePool(Node, &FreePtr);
    printAll(AList, FreePtr, Node);
    CreateList(&AList);          //DHMIOYRGIA LISTAS

    printf("DWSE ARI8MO GIA EISAGWGH STH LISTA: ");
    scanf("%d", &AnItem);
    PredPtr=NilValue;
    Insert(&AList, Node,&FreePtr, PredPtr, AnItem); //EI

    TraverseLinked(AList, Node);      //DIASXISH LISTAS
    printAll(AList, FreePtr, Node);

    return 0;
}
```

Search(...) → -1

```
void Insert(ListPointer *List, NodeType Node[],ListPointer *FreePtr, ListPointer PredPtr, ListElementType Item)
{
  ListPointer TempPtr;
  GetNode(&TempPtr,FreePtr,Node);
  if (!FullLList(TempPtr)) {
    if (PredPtr==NilValue)
    {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =*List;
        *List =TempPtr;
    }
    else
      {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =Node[PredPtr].Next;
        Node[PredPtr].Next =TempPtr;
      }
}
  else
    printf("Full List ...\n");
}
```

TempPtr = 1, FreePtr = 2

Το 9 δεν έχει προηγούμενο

Σ. Λίστα ταξινομημένη

NumberOfNodes

5

```
#define NilValue -1

typedef int ListElementType;

typedef int ListPointer;

typedef struct {
    ListElementType Data;
    ListPointer  Next;
} NodeType;
```

*List

1

Node[ ]

	Data	Next
0	12	~~12~~
1	9	0
2	31	~~3~~ -1
3	-1	4
4	-1	-1

FreePtr

~~2~~3

Item

31

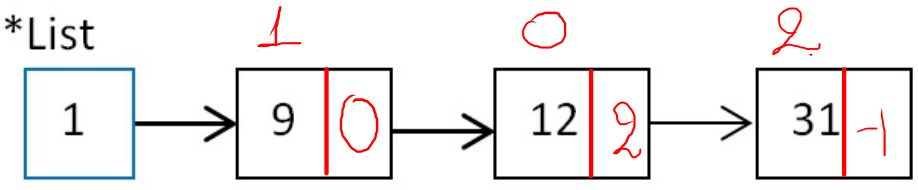
TempPtr

2

PredPtr

0

*List




```
int main()
{
    ListPointer AList;
    NodeType Node[NumberOfNodes];
    ListPointer FreePtr,PredPtr;
    ListElementType AnItem;

    InitializeStoragePool(Node, &FreePtr);
    printAll(AList, FreePtr, Node);
    CreateList(&AList);           //DHMIOYRGIA LISTAS

    printf("DWSE ARI8MO GIA EISAGWGH STH LISTA: ");
    scanf("%d", &AnItem);
    PredPtr=NilValue;
    Insert(&AList, Node,&FreePtr, PredPtr, AnItem); //EI

    TraverseLinked(AList, Node);      //DIASXISH LISTAS
    printAll(AList, FreePtr, Node);

    return 0;
}
```

Search(...) → 0

```
void Insert(ListPointer *List, NodeType Node[],ListPointer *FreePtr, ListPointer PredPtr, ListElementType Item)
{
  ListPointer TempPtr;
  GetNode(&TempPtr,FreePtr,Node);
  if (!FullLList(TempPtr)) {
    if (PredPtr==NilValue)
    {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =*List;
        *List =TempPtr;
    }
    else
      {
        Node[TempPtr].Data =Item;
        Node[TempPtr].Next =Node[PredPtr].Next;
        Node[PredPtr].Next =TempPtr;
      }
}
  else
    printf("Full List ...\n");
}
```

TempPtr = 2, FreePtr = 3

Το 31 έχει προχωρήσει το 12