



Sat	Sun	Mon	Tue	Wed	Thu	Fri
-----	-----	-----	-----	-----	-----	-----

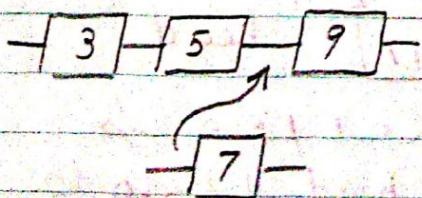
Revision

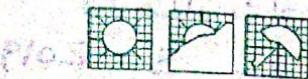
Q1 : Inorder static Linear List insert Function

```

bool List :: insert (const data ElemtType & e)
{ if ( no of items >= max size list)
    return false;
else if( no of items == 0) → خاصية list
    { List [0] = e; no of items ++; } بدخل او اول دخل elem
else { for (i=0; i< no of items; i++)
    { if ( List array [i] >= e) → (1)
        for(j=(no of items -1); j<=i; j--)
        { List [j+1] = List [j]; }
        break; }
    List [i] = e;
    no of items++;
    return true;
}
    
```

(1) الجزء ده عايز ادخل elem في list فارى elem وقارنها بالـ list كل بتابعه اكبر مفخر ماشى لـ elem او اصغر من elem وساعتها اتحط قبل عامله وازود عدد الـ items بواحد





Sat	Sun	Mon	Tue	Wed	Thu	Fri
-----	-----	-----	-----	-----	-----	-----

Page _____

Date / /

Q2:



لـ 5، left N rotation (لـ 5، عايز نـ 1، left List Line)



List :: ROL (void)

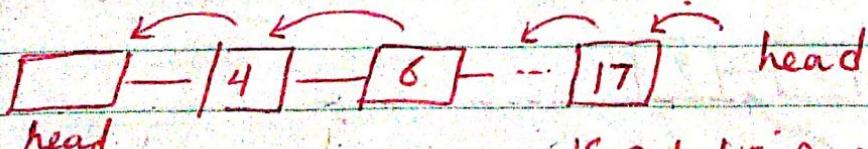
```

{ Link ptr;
List ELEMType temp;
ptr = head;
temp = head->item;
if (ptr == Null)
    return;
while (ptr->next != Null) {
    ptr->item = ptr->next->item;
    ptr = ptr->next;
    ptr->item = temp;
}
  
```

لـ 5، left N rotation (لـ 5، عايز نـ 1، left N rotation)

on elem (لـ 5، عايز نـ 1، Head) or left N elem (لـ 5، عايز N shift)

left N elem (لـ 5، عايز N shift)



is right و save خافظ على Temp (لـ 5، عايز N shift)

ptr->next != Null (لـ 5، عايز N shift)

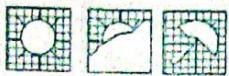
17 (لـ 5، عايز N shift)

ptr->next != Null (لـ 5، عايز N shift)

Temp (لـ 5، عايز N shift)

17 (لـ 5، عايز N shift)

Amon20M



Sat	Sun	Mon	Tue	Wed	Thu	Fri
-----	-----	-----	-----	-----	-----	-----

Page _____

Date _____

Another solution (without while loop)

List::Rot(Void)

```
{ assert( !emptyList() );
tail->next = head;
tail = tail->next;
head = head->next;
tail->next = Null; }
```

7

فكرة الكود اني بـ 7 خطوات ابرم list ما احرك tail او head
Shift left كلac jis tail il 1st

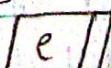
Q3: Insert after (Const ListElemType &e, Link p)
insert a node containing e after a node pointed to by p

Void insert after (Const ListElemType &e, Link p)

```
{ Link addedNode; assert( !emptyList() );
addedNode = New Node;
addedNode->item = e;
addedNode->next = p->next;
p->next = addedNode; }
```

}

فكرة الكود اني خالص هرجع المايل من list assert il *
فاتخذه هتخرج من الـ assert il *
لأنها معرفة قبلياً Node فـ * تكون في



فكرة الكود ياعواحد في الـ Node il Create JSD (1)

ـ P->next ياعـ P مخـ يـ next il (2)

ـ P->next ياعـ P مخـ يـ next il

ـ P->next ياعـ P مخـ يـ next il (3)

P

Aman2M



The rest of Q.3.

Insert Before (const ListType & e, Link p)

insert a node containing e before a node pointed to by p

insert before (const ListType & e, Link p) {

insert after (const ListType & e, Link p) {

ListElemType Temp;

Temp = p->next->item;

p->next->item = p->item;

p->item = Temp;

} // end of insert before

فكرة الكود هي إنشاء临时节点 temp لـ e، ثم تبديل指针 p->next->item بـ temp->item، وتحل temp->item بـ p->item.

بعد ذلك، يُرجع temp->item إلى e، وتحل temp->item بـ p->item.

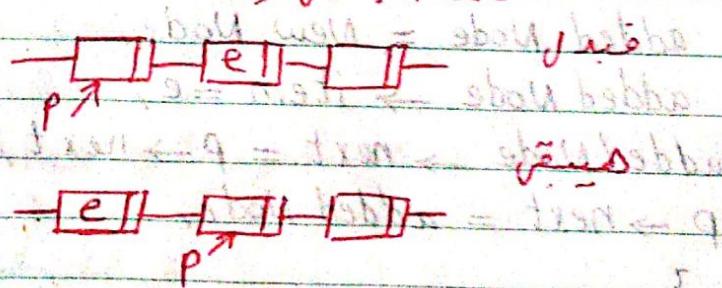
استخراج insert before و insert after:

insert before (const ListType & e, Link p) {

 insert after (const ListType & e, Link p); // (1)

 swap (e, p->item); // (2)

 insert after (const ListType & e, Link p); // (3)



ملاحظة:

في الـ insert after ، هو الـ tail الذي ينبع من الـ insert after.

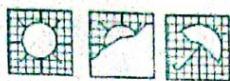
وهو المعنون بـ p->next ، وهو المعنون بـ tail .

حيث أن p == tail ، فيجب التأكد من أن p != null .

لو في الـ main الكائن الذي قيل له define class ، فيكتب

الـ class بالطريقة التالية: class Variables {

 Variables(); // constructor



Sat	Sun	Mon	Tue	Wed	Thu	Fri
-----	-----	-----	-----	-----	-----	-----

Page

Date

Q4:

List* Front Back Split: member Function of List

splits List into two parts one for the first half
and the second half is returned to me.

الفكرة هنا أننا ن Sunder list كبيرة عايز أعملها

وأخذ النص الثاني منها

فلو مثلاً الـ list مقسموا كذا

{ 17, 12, 7, 5, 4, 23 }

{ 17, 12, 7 }

First half is
saved as old list

{ 5, 4, 23 }

Second half is

returned by a function.

{ 17, 12, 5, 4, 23 }

{ 17, 12, 5 }

First half is
saved as old list

{ 4, 23 }

Second half is

list returned to me

فكرة الكود المحرر في الصفحة الجایة

لابياعز رضي الـ list الثاني فالاول هقوله عايز مكأن

الـ list الجديدة واعمل assert بعدها وعشان أقسم

الـ list لذين لازم الاول أعرف عددها فمحمل loop بتعريفه

كم node فيها الـ list الـ list دايمه وتخزينه في ا

لو الرقم اللي معندي odd هزروا واحد واقسموا على 2

ترى (1) (2) لكن لو الرقم ده even مقصمو على 2 عادلوا (1)(2) بس

المهم دلوقتي أنا عايز رضي الـ list الثاني فمحمل loop تاف كده

نفس الـ list فكره هتوصيله للأول Node هستخرج منه الـ list الجديدة

وهبيق هوده الـ head بتاع الـ list الجديدة والـ Node اللي

قىله هوار tail بتاع نفس الـ list الأول اللي وار tail

باتاع الـ list الأصلية هبيق الـ tail الـ list الجديدة

Amr2016



Sat Sun Mon Tue Wed Thu Fri

Page _____

Date _____

List * List :: Front Back Split () {

 List * part2 = New List;

 Assert (part2 != NULL); otherwise illegal

 int i=0; because initially head == alt ?

 Link temp = head;

 while (temp != NULL) {

 i++;

 temp = temp -> next; }

 if (i%2 != 0)

 i++;

 ← (1) step 3

 i = i/2; temp = head; ← (2) step 4

 for (int j=0; j < i; j++)

 temp = temp -> next;

 Link temptail = tail;

 tail = temp;

 tail -> next = NULL;

 part2 -> head = temp -> next;

 part2 -> tail = temptail;

 return part2;

}

Q5: member Function inter change

(loops are valid and they are needed here)

i/p [7] [13] [5] [8] --- [11] [3]

o/p [13] [7] [8] [5] --- [3] [11x]

Aman2011



عایز آبیل بین کل اتسنن nodes و بعض ری الک کانت
خن ال Assignment ← shuffle

```

int List :: interchange ()
{
    link node;
    ListElemType x;
    int count = 0;
    if (head == NULL || head->next == NULL)
        return count;
    else
    {
        node = head;
        while (node != NULL && node->next != NULL)
        {
            x = node->item;
            node->item = node->next->item;
            node->next->item = x;
            count++;
            node = node->next->next;
        }
        return count;
    }
}

```

هذا : `Live Count` Function لطلب في السؤال أن تترجم عدد المدخلات التي تم تبديلها بـ `int` ولكن لو مطابق ستغير `Void Function` عادي . مكرة الأكود :

(1) لوار list فاضية او فيغا elem واحد من حرف اطبعه او
 interchange

(2) لوار list لام فيغا elements فكتور من 1 فيغا بعدي
 swap بين elem و head of loop باعدها

الآن جنبه غير جيد $13 \rightarrow 7$ $7 \rightarrow 13$ $5 \rightarrow 8$ $8 \rightarrow 5$
لذلك ن Swap المنشئين و لكننا $5 \rightarrow 8$ $8 \rightarrow 5$