

How to replace the items in lists after calculating each items between two lists.

**Error: (on screen)**

**List index smaller than 1**

**Replace list item: Attempt to replace item number 0 of the list (5 3 5). The minimum valid item number is 1.**

For example:

There are two lists as followed: ( 3 items in each list )

=====

Gain List : ( -3 -4 -6 )

Pay List : ( 5 3 5 )

=====

Then I need to calculate the items each other (between two lists) with order and replace the value in the original position (index).

The items (value) in two lists will finally be 0.

And I will display some text on my screen between every round.

---

**\* We don't calculate the item which value is 0.**

**First round:**

Index	1	2	3
Gain (List)	<del>-3</del> 0	-4	-6
Pay (List)	<del>-5</del> 2	3	5

Calculate:      Gain (index) + Pay (index)  
 Gain (1) + Pay (1)      (|Gain| < Pay)  
 = -3 + 5 = 2

Do in the same time: (1) Set Next Gain (1) = 0  
 (2) Set Next Pay (1) = Gain (1) + Pay (1)  
 = -3+5 = 2

=====

**Second round:**

New lists

Gain List : ( 0 -4 -6 )

Pay List : ( 2 3 5 )

Index	1	2	3
Gain (List)	0	<del>-4</del> 2	-6
Pay (List)	<del>-2</del> 0	3	5

Calculate:      Gain (index) + Pay (index)  
 Gain (2) + Pay (1)      (|Gain| > Pay)  
 = -4 + 2 = -2

Do in the same time: (1) Set Next Pay (1) = 0  
 (2) Set Next Gain (2) = Gain (2) + Pay (1)  
 = -4 + 2 = -2

### Third round:

New lists

Gain List : ( 0 -2 -6 )

Pay List : ( 0 3 5 )

Index	1	2	3
Gain (List)	0	<del>-2</del> 0	-6
Pay List)	0	<del>3</del> 1	5

Calculate: Gain (index) + Pay (index)

$$\text{Gain (2) + Pay (2)} \quad (|\text{Gain}| < \text{Pay})$$

$$= -2 + 3 = 1$$

Do in the same time: (1) Set Next Gain (2) = 0

(2) Set Next Pay (2) = Gain (2) + Pay (2)

$$= -2 + 3 = 1$$

---

---

### Fourth round:

New lists

Gain List : ( 0 0 -6 )

Pay List : ( 0 1 5 )

Index	1	2	3
Gain (List)	0	0	<del>-6</del> -5
Pay List)	0	<del>1</del> 0	5

Calculate: Gain (index) + Pay (index)

$$\text{Gain (3) + Pay (2)} \quad (|\text{Gain}| > \text{Pay})$$

$$= -6 + 1 = -5$$

Do in the same time: (1) Set Next Pay (2) = 0

(2) Set Next Gain (3) = Gain (3) + Pay (2)

$$= -6 + 1 = -5$$

---

---

**Fifth round:**

New lists

Gain List : ( 0 0 -5 )

Pay List : (0 0 5)

Index	1	2	3
Gain (List)	0	0	<del>-5</del> 0
Pay List)	0	0	<del>5</del> 0

Calculate: Gain (index) + Pay (index)

$$\text{Gain (3) + Pay (3)} \quad (|\text{Gain}| > \text{Pay})$$

$$= -5 + 5 = 0$$

Do in the same time: (1) Set Next Pay (3) = Gain (3) + Pay (3)

$$= 0$$

(2) Set Next Gain (3) = Gain (3) + Pay (3)

$$= 0$$

---

---

# App Inventor 2

Display hidden components in Viewer

The screenshot displays the App Inventor 2 interface. On the left is the mobile app viewer, and on the right is the component palette. Red arrows indicate the mapping between components in the palette and their placement in the app viewer.

**Viewer Components:**

- Screen1 (Header)
- Gain list: Text for Label2
- Pay list: Text for Label3
- Test\_List\_Gain (ListView)
- Text\_List\_Pay (Text)
- Calculate (Button)

**Component Palette:**

- Screen1
  - HorizontalArrangement1
    - Label1
    - Label\_gain\_list
  - HorizontalArrangement2
    - Label4
    - Label\_pay\_list
  - HorizontalArrangement3
    - Label5
    - ListView\_test\_list\_gair
  - HorizontalArrangement4
    - Label6
    - ListView\_test\_list\_Pay
  - Button1

**Red Arrows:**

- From Label\_gain\_list to Gain list: Text for Label2
- From Label\_pay\_list to Pay list: Text for Label3
- From ListView\_test\_list\_gair to Test\_List\_Gain
- From Label6 to Text\_List\_Pay
- From Button1 to Calculate

## ➤ Variables

initialize global Gain1 to -3

initialize global Gain2 to -4

initialize global Gain3 to -6

initialize global Gain\_List to create empty list

initialize global gain\_for\_register to 0

initialize global pay\_for\_register to 0

initialize global Pay1 to 5

initialize global Pay2 to 3

initialize global Pay3 to 5

initialize global Pay\_List to create empty list

initialize global gain\_for\_index to 0

initialize global pay\_for\_index to 0

initialize global Result\_for\_Cal\_loop to 0

initialize global test\_list\_for\_gain to create empty list

initialize global test\_list\_for\_pay to create empty list

```

when Button1 Click
do
  add items to list list
  item get global Gain_List
  item get global Gain1
  item get global Gain2
  item get global Gain3
  add items to list list
  item get global Pay_List
  item get global Pay1
  item get global Pay2
  item get global Pay3

  set Label_gain_list Text to get global Gain_List
  set Label_pay_list Text to get global Pay_List

  for each Gain in list get global Gain_List
  do
    for each Pay in list get global Pay_List
    do
      if
        get Gain ≠ 0 and get Pay ≠ 0
      then
        if
          absolute get Gain > get Pay
        then
          set global gain_for_index to 0
          set global pay_for_index to 0
          set global gain_for_register to 0
          set global pay_for_register to 0
          set global gain_for_register to get Gain
          set global pay_for_register to get Pay
          set global gain_for_index to index in list thing get Gain list get global Gain_List
          set global pay_for_index to index in list thing get Pay list get global Gain_List

          replace list item list get global Gain_List
          index get global gain_for_index
          replacement call Calculate_Gain_Pay
            Cal_Gain get global gain_for_register
            Cal_Pay get global pay_for_register

          replace list item list get global Pay_List
          index get global pay_for_index
          replacement 0

          append to list list1 get global test_list_for_gain
          list2 get global Gain_List
          set ListView_test_list_gain Elements to get global test_list_for_gain
          append to list list1 get global test_list_for_pay
          list2 get global Pay_List
          set ListView_test_list_Pay Elements to get global test_list_for_pay
        end if
      end if
    end do
  end do
end do

```

Build the lists.  
 Gain\_List =(Gain1 Gain2 Gain3)  
 Pay List = (Pay1 Pay2 Pay3)

If the Gain item = 0 "OR" Pay item = 0  
 We don't calculate the item.

Call Calculate\_Gain\_Pay module to  
 calculate the value then be the item of  
 replacement item.

Replace the item by 0.  
 Because Pay item < |Gain item|

Calculate Module

```

to Calculate_Gain_Pay Cal_Gain Cal_Pay
result
do
  set global Result_for_Cal_loop to 0
  set global Result_for_Cal_loop to
  [
    get Cal_Gain + get Cal_Pay
  ]
  result get global Result_for_Cal_loop
end do

```

```

else if [absolute] [get Gain] < [get Pay]
then
  set global gain_for_index to 0
  set global pay_for_index to 0
  set global gain_for_register to 0
  set global pay_for_register to 0
  set global gain_for_register to [get Gain]
  set global pay_for_register to [get Pay]
  set global gain_for_index to [index in list thing] [get Gain] [list] [get global Gain_List]
  set global pay_for_index to [index in list thing] [get Pay] [list] [get global Gain_List]
  set global gain_for_register to [get Gain]
  replace list item list [get global Gain_List]
  index [get global gain_for_index]
  replacement 0
  replace list item list [get global Pay_List]
  index [get global pay_for_index]
  replacement call [Calculate_Gain_Pay]
  Cal_Gain [get global gain_for_register]
  Cal_Pay [get global pay_for_register]
  append to list list1 [get global test_list_for_gain]
  list2 [get global Gain_List]
  set [ListView_test_list_gain] . Elements to [get global test_list_for_gain]
  append to list list1 [get global test_list_for_pay]
  list2 [get global Pay_List]
  set [ListView_test_list_Pay] . Elements to [get global test_list_for_pay]

```

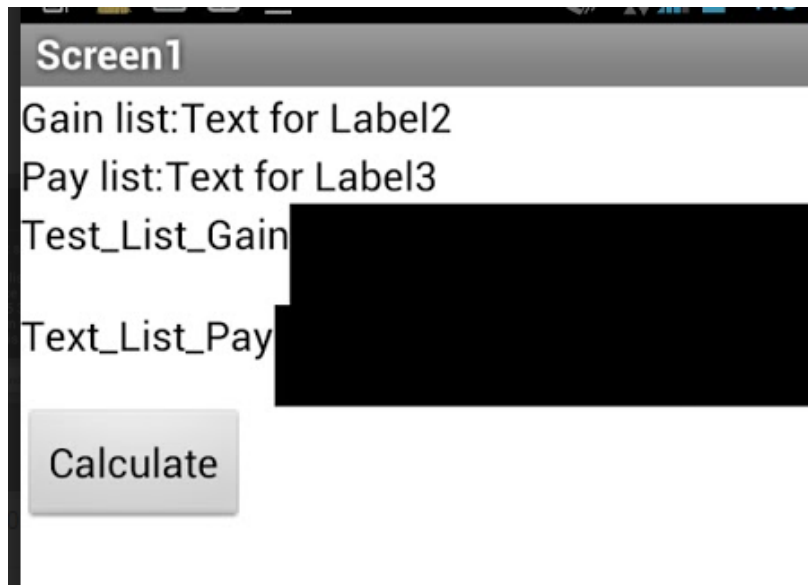
```

else if [absolute] [get Gain] = [get Pay]
then
  set global gain_for_register to 0
  set global pay_for_register to 0
  set global gain_for_index to 0
  set global pay_for_index to 0
  set global gain_for_register to [get Gain]
  set global pay_for_register to [get Pay]
  set global gain_for_index to [index in list thing] [get Gain] [list] [get global Gain_List]
  set global pay_for_index to [index in list thing] [get Pay] [list] [get global Gain_List]
  replace list item list [get global Gain_List]
  index [get global gain_for_index]
  replacement call [Calculate_Gain_Pay]
  Cal_Gain [get global gain_for_register]
  Cal_Pay [get global pay_for_register]
  replace list item list [get global Pay_List]
  index [get global pay_for_index]
  replacement call [Calculate_Gain_Pay]
  Cal_Gain [get global gain_for_register]
  Cal_Pay [get global pay_for_register]
  append to list list1 [get global test_list_for_gain]
  list2 [get global Gain_List]
  set [ListView_test_list_gain] . Elements to [get global test_list_for_gain]
  append to list list1 [get global test_list_for_pay]
  list2 [get global Pay_List]
  set [ListView_test_list_Pay] . Elements to [get global test_list_for_pay]

```



➤ The screen of my Screen



Then press "Calculate" Button

Error shows up !

