CSC530, Data Structures, instructor: Dr. Zhen Jiang
Project 1 assignment (Part B – Event control loop: 12x3 = 36 points)

Submit via D2L. Late submissions are not acceptable.

Evaluation: You need to answer all questions and finish each step. Make the answers consistent with your program. You must also provide your code. You will loss all the points if no complete program is given and consistent all your design steps. Array CANNOT be used.

Sample in class: Calculate the result 1+3+5+7+…+99

Step 1 (1 pt): What’s the repetition body you found (explain it in your own words):
addition

Step 2a (1 pt): What do you expect the computer do at the first iteration
(or what’s the format of this body when you first repeat it):
1+3

Step 2b (1 pt): What do you expect the computer do at the second iteration. If necessary, rewrite the previous iteration.
total = 1 + 3
total = total + 5

Step 2c (1 pt): What do you expect the computer do at the third iteration. If necessary, rewrite the previous iteration.
total = total + 7

Step 3a (1 pt): What do you expect the computer do at the jth iteration
total = total + 2*j+1 // to make sure the entire program is possible to develop

Step 3b (1 pt): What’s general format of this repetition body for all above repeated steps?
Total = total + □

Step 4 (0 pt): We know it is event control loop!

Step 5b (1 pt): In what situation, the loop is going to stop. □>99

Step 6b (1 pt): What is the condition (part) in loop program? □<=99 //usually it is opposite of the above

Step 7b (1 pt): What is the range of values to keep the condition true? Show them all. □ = 3, 5, ..., 99

Step 8b (1 pt): What is the initialized value (the value for checking the condition at the first time)? □ = 3

Step 9b (1 pt): How to change these values, from the previous one to the next one, in a general format?
□ = □ + 2

Step 10b (1 pt): What is the initialization?
total = 1

Final verification step (0): What’s the program.
Lack of the program will be charged all the points in the above steps!
You MUST provide the program to convince the grader that you follow the exact procedure to develop the code. Before your submission, check if there is any inconsistence, such as different initialization value or different number of times to repeat the iteration body. Anything missed here will lead to a charge in the above step.)
total = 1;
c = 3;
while (c <= 99) {
total = total + c;
c = c + 2;
}

Lack of the program will be charged all the points in the above steps!
1. The world population reached 6 billion people in 1999 and was growing at the rate of 1.4% each year. Write a program to determine when the population will exceed 10 billion. Do all the steps in the above sample.

2. A TV set is purchased with a loan of $563 to be paid off with monthly payment (no more than $116). The interest rate is 1% per month. Display the amount of last bill. Do all the steps in question 1.

3. Calculate the result $1+2+4+8+16+32+\ldots+k$, where $k$ is the first number larger than 4000. Do all the steps in question 1. Note that your program should also work when 4000 changes to another number.

Followings are some samples that explain the necessity of using this 13-step development procedure.

Is this a counter-control loop?

What is the first iteration?

```
total = 0 + 1
```

What is the second iteration?

```
total = total + 2!
```

What is the third iteration?

```
total = total + 4!
```

What is the general format?

```
Total = total + □
```

Does it make sense?

(a) in what situation, the loop is going to stop.

```
□ >= 100 (matched)
```

(b) What is the range of values to keep the condition true?

```
0, 2, 4, 8, \ldots, 98! does not match, what’s wrong?
```

(c) How to change these values, from the previous one to the next one, in a general format?

```
□ = □ + 2 (Matched)
```

(d) What is the initialization?

```
i = 0 (□ = 0)
```

So, change the condition to $i = 1(□ = 1)$ and then verify the entire process again.
Is this a counter-control loop?

What is the first iteration?

total = 3 + 1

What is the second iteration?

total = total + 3 (conflict with the previous one)

What is the third iteration?

total = total + 5

What is the general format?

Total = total + □

Does it make sense?

(c) in what situation, the loop is going to stop.

□ >= 100 (matched)

(d) What is the range of values to keep the condition true?

1, 3, 5, 7, …, 99 (matched)

(c) How to change these values, from the previous one to the next one, in a general format?

□ = □ + 2 (Matched)

(d) What is the initialization?

total = 3 ! (Does not match)

So, change the initialization to total = 0 and then verify the entire process again.
Is this a counter-control loop?

What is the first iteration?
\[ \text{total} = 1+3 \]

What is the second iteration?
\[ \text{total} = \text{total} + 5 \]

What is the third iteration?
\[ \text{total} = \text{total} + 7 \]

What is the general format?
\[ \text{Total} = \text{total} + □ \]

Does it make sense?
  (e) in what situation, the loop is going to stop.
    \[ □ \geq 99 \] (does not match!)
  
  (f) What is the range of values to keep the condition true?
    \[ 3, 5, 7, \ldots, 97 \] (does not match, what’s wrong?)

(c) How to change these values, from the previous one to the next one, in a general format?
\[ □ = □ + 2 \] (Matched)

(d) What is the initialization?
\[ □ = 3 \] (matched)

So, change the condition to \( i \leq 99 \) and then verify the entire process.

similar!
Change initialization to \( \text{total} = 1 \)
Is this a counter-control loop?

What is the first iteration?
*total = 1+2 (does not match)*

What is the second iteration?
*total = total +4*

What is the third iteration?
*total = total +6*

What is the general format?
*Total = total + □*

Does it make sense?

(g) in what situation, the loop is going to stop.
*□ >= 101 (stop at 99 or 100, and is a good condition)*

(h) What is the range of values to keep the condition true?
*2, 4, 6, …, 100 ! (does not match, what’s wrong?)*

(c) How to change these values, from the previous one to the next one, in a general format?

□ = □ + 2 (Matched)

(d) What is the initialization?
*i = 0 (□ = 2 does not match!)*

So, change the initialization to *i = 1 (□ = 3 )* and then verify the entire process.

(similar)
Change to *i = -1* may cause problem if *i* is declared as char!
Is this a counter-control loop?
No! because the program does not have the right number of body repetition and the whole program cannot be fixed by correcting such a number (more than one error if you insist it is a counter control loop).

What is the first iteration?
\[ \text{total} = 0 + 1 \]

What is the second iteration?
\[ \text{total} = \text{total} + 2 \]

What is the third iteration?
\[ \text{total} = \text{total} + 3 \]

What is the general format?
\[ \text{Total} = \text{total} + □ \]

Does it make sense?
(i) in what situation, the loop is going to stop.
\[ □ \geq 100 \text{ (stop at 99)} \]

(j) What is the range of values to keep the condition true?
1, 2, 3, …, 99! (does not match, what’s wrong?)

(c) How to change these values, from the previous one to the next one, in a general format?
\[ □ = □ + 1 \text{ (Does not match!)} \]

(d) What is the initialization?
\[ i = 0 \text{ (} □ = 1 \text{ could match)} \]

So, change the increment to \( i = i + 2 \) and then verify the entire process.

Is this a counter-control loop?

What is the first iteration?
\[ \text{total} = 1 \]

What is the second iteration?
\[ \text{total} = \text{total} + 3 \]

What is the third iteration?
\[ \text{total} = \text{total} + 5 \]

What is the general format?
\[ \text{Total} = □ \text{ (does not match)} \]

Does it make sense?
(k) in what situation, the loop is going to stop.
\[ □ \geq 100 \text{ (stop at 99)} \]

(l) What is the range of values to keep the condition true?
1, 3, 5, …, 99 (matched)

(c) How to change these values, from the previous one to the next one, in a general format?
\[ □ = □ + 2 \text{ (matched)} \]

(d) What is the initialization?
\[ i = 1, \text{total} = 0 \text{ (could match)} \]

So, change the general format to \( \text{total} = \text{total} + i \) and then verify the entire process.
Is this a counter-control loop?
No! Because \( i = 1 \) and change to \( i = 0 \) is not enough!

What is the first iteration?
\[
\text{total } = 0 + 1
\]

What is the second iteration?
\[
\text{total } = \text{total } + 3
\]

What is the third iteration?
\[
\text{total } = \text{total } + 5
\]

What is the general format?
\[
\text{Total } = \text{total } + \Box
\]

Does it make sense?
(o) in what situation, the loop is going to stop.
\[ i \geq 100 (\Box \geq 199) \]

(p) What is the range of values to keep the condition true?
1, 2, 3, …, 99 (\( \Box = 99 \) matched!) Does not match!

(c) How to change these values, from the previous one to the next one, in a general format?
\[ \Box = \Box + 2 (i = i + 1) \] matched

(d) What is the initialization?
\[ i = 1, \text{total} = 0 \] (could match)

So, change the condition to \( i \leq 50 \) and then verify the entire process.
Is this a counter-control loop?
No! Because \( i = 1 \) and change to \( i = 0 \) is not enough!

What is the first iteration?
\[
\text{total} = 1 + 8 \quad (! 3+5)
\]

What is the second iteration?
\[
\text{total} = \text{total} + 16 \quad (! 7+9)
\]

What is the third iteration?
\[
\text{total} = \text{total} + 24 \quad (! 11+13)
\]

What is the general format?
\[
\text{Total} = \text{total} + \Box
\]

Does it make sense?

(q) in what situation, the loop is going to stop.
\[
i >=25 \quad (\Box >= 200)
\]

(r) What is the range of values to keep the condition true?
\[
1, 2, 3, \ldots, 24 \quad (\Box = 192 = 95+97!)
\]

(c) How to change these values, from the previous one to the next one, in a general format?
\[
\Box = \Box + 8 \quad (i = i + 1) \quad \text{(matched)}
\]

(d) What is the initialization?
\[
i = 1, \text{total} = 1 \quad \text{(matched)}
\]

Nothing wrong? How about +99? So, add \( \text{total} += 99 \) at the end.
Is this a counter-control loop?
No! Because \( i = 1 \) and change to \( i = 0 \) is not enough!

What is the first iteration?
\[
\text{total} = 1 + 12 \, (! \, 5+7)
\]

What is the second iteration?
\[
\text{total} = \text{total} + 20 \, (! \, 9+11)
\]

What is the third iteration?
\[
\text{total} = \text{total} + 28 \, (! \, 13+15)
\]

What is the general format?
\[
\text{Total} = \text{total} + □
\]

Does it make sense?

(s) in what situation, the loop is going to stop. \( i \geq 25 \, (! \, □ \geq 204) \)

(t) What is the range of values to keep the condition true? 1, 2, 3, \ldots, 24 (\( □ = 196 = 97+99! \))

(c) How to change these values, from the previous one to the next one, in a general format?
\[
□ = □ + 8 \, (i = i + 1) \, (\text{matched})
\]

(d) What is the initialization?
Change the initialization to \( \text{total} = 4 \) (not \( i = 4! \))

Is this a counter-control loop?
No! Because \( i = 4 \) and \( i = i + 8! \)

What is the first iteration?
\[
\text{total} = 0 + 4 \, (! \, 1+3)
\]

What is the second iteration?
\[
\text{total} = \text{total} + 12 \, (! \, 5+7)
\]

What is the third iteration?
\[
\text{total} = \text{total} + 20 \, (! \, 9+11)
\]

What is the general format?
\[
\text{Total} = \text{total} + □
\]

Does it make sense?

(u) in what situation, the loop is going to stop. \( i \geq 25 \, (! \, □ \geq 25) \)

(v) What is the range of values to keep the condition true? 4, 12, 20 (\( □ = 9+11! \))

(c) How to change these values, from the previous one to the next one, in a general format?
\[
□ = □ + 8 \, (i = i + 8) \, (\text{matched})
\]

(d) What is the initialization?
\[
i = 4, \, \text{total} = 0 \, (\text{matched})
\]
Change the condition to \( i \leq 196. \)
Is this a counter-control loop?
No! Because $i = 1$ and $i = i + 8!$

What is the first iteration?
$\text{total} = 0 + 1$

What is the second iteration?
$\text{total} = \text{total} + 9 (! 5+7 - 3, \text{not } 3+5+1 \text{ because of } 97+99)$

What is the third iteration?
$\text{total} = \text{total} + 17 (! 9+11 - 3 )$

What is the general format?
$\text{Total} = \text{total} \ + \ □$

Does it make sense?
(w) in what situation, the loop is going to stop.
\[ i > 196 (! > 196) \]

(x) What is the range of values to keep the condition true?
\[ 1, 9, 17, …, 193 (! = 97+99 - 3!) \]

(c) How to change these values, from the previous one to the next one, in a general format?
\[ □ = □ + 8 (i = i + 8) \text{ (matched )} \]

(d) What is the initialization?
Change the initialization $o i = 1+3$.

---

Is this a counter-control loop?
No! Because $i = 4$ and $i = i + 8!$

What is the first iteration?
$\text{total} = 1 + 4 (1+3+1 \text{ because of } 97+99)$

What is the second iteration?
$\text{total} = \text{total} + 12 (! 5+7)$

What is the third iteration?
$\text{total} = \text{total} + 20 (! 9+11 )$

What is the general format?
$\text{Total} = \text{total} \ + \ □$

Does it make sense?
(y) in what situation, the loop is going to stop.
\[ i > 196 (! > 196) \]

(z) What is the range of values to keep the condition true?
\[ 4, 12, 20, …, 196 (! = 97+99!) \]

(c) How to change these values, from the previous one to the next one, in a general format?
\[ □ = □ + 8 (i = i + 8) \text{ (matched )} \]

(d) What is the initialization?
Change the initialization $o \text{total} = 0$. 