

گزارش کار تمرین کامپیوتری چهارم آزمون نرم افزار

مجید فریدفر 810199569

اولدوز نیساری 810199505

لینک ریپازیتوری: <https://github.com/maj2idfar/Baloot>

هش آخرین کامیت: 5bc9048

سوال اول

1. لیست clause ها :

b , a, c

2.

Row num	کل عبارت	$a \wedge \neg c$	$\neg a \wedge \neg c$	$a \wedge \neg b$	c	b	a
1	F	F	F	F	T	T	T
2	T	T	F	F	F	T	T
3	T	F	F	T	T	F	T
4	T	T	F	T	F	F	T
5	F	F	F	F	T	T	F
6	T	F	T	F	F	T	F
7	F	F	F	F	T	F	F
8	T	F	T	F	F	F	F

3.

GACC:

Major clause = "a" :

Row pairs : {(3,7)}

Major clause = "b":

Row pairs : {(1,3)}

Major clause = "c":

Row pairs : { (1,2) , (1,6) , (1,8) , (2,5) , (2,7), (5,6), (5,8) , (6,7) , (7,8)}

.4

RACC :

a) Major clause = "a" :

Row pairs : {(3,7)}

b)Major clause = "b":

Row pairs : {(1,3)}

c) Major clause = "c":

Row pairs : { (1,2) ,(5,6), (7,8)}

.5

RICC :

a) Major Clause = "a" :

Row pairs = {(1,5),(2,6),(4,8)}

b) Major Clause = "b":

Row pairs = {(2,4),(5,7), (6,8)}

c) Major Clause = "c":

Row pairs = {3,4}

سوال 2

1) Quantity :

برای این ورودی 4 حالت مختلف می توان در نظر گرفت :

- مقدار های کمتر از 0 : Q1
- مقدار 0 : Q2
- مقدار 1 : Q3
- مقدار های بزرگتر از 1 : Q4

2) Unit_price :

برای این ورودی 3 حالت مختلف می توان در نظر گرفت.

- مقدار های کمتر از 0 : U1
- مقدار برابر 0 : U2
- مقدار های بزرگتر از 0 : U3

3) Discount :

برای این ورودی 5 حالت مختلف می توان در نظر گرفت.

- مقدار های کمتر از 0 : D1
- مقدار برابر 0 : D2
- مقدار بین 0 و 1 : D3
- مقدار برابر 1 : D4
- مقدار های بزرگتر از 1 : D5

حال تست کیس های مناسب طراحی می کنیم :

1. Q1,U1,D1
2. Q2,U2,D1

3. Q3,U3,D1
4. Q4,U1,D1
5. Q1,U3,D2
6. Q2,U1,D2
7. Q3,U2,D2
8. Q4,U3,D2
9. Q1,U2,D3
10. Q2,U3,D3
11. Q3,U1,D3
12. Q4,U3,D3
13. Q1,U1,D4
14. Q2,U2,D4
15. Q3,U3,D4
16. Q4,U3,D4
17. Q1,U1,D5
18. Q2,U2,D5
19. Q3,U3,D5
20. Q4,U2,D5

حال برای هر یک از تست ها به ترتیب assert ها ی زیر را انجام می دهیم :

- 1 – assertEquals("Invalid input", calculate_total_cost(-5, -4, -1))
- 2 – assertEquals("Invalid input", calculate_total_cost(0, 0, -1))
- 3 – assertEquals("Invalid input", calculate_total_cost(1, 4, -1))
- 4 – assertEquals("Invalid input", calculate_total_cost(5, -4,-1))
- 5 – assertEquals("Invalid input", calculate_total_cost(-5, 4, 0))
- 6 – assertEquals("Invalid input", calculate_total_cost(0, -4, 0))
- 7 – assertEquals("Invalid input", calculate_total_cost(1, 0, 0))

8 – assertEquals(4, calculate_total_cost(5, 4, 0))
9 – assertEquals("Invalid input", calculate_total_cost(-5, 0, 0.5))
10 – assertEquals("Invalid input", calculate_total_cost(0, 4, 0.5))
11 – assertEquals("Invalid input", calculate_total_cost(1, -4, 0.5))
12 – assertEquals(1, calculate_total_cost(5, 4, 0.5))
13 – assertEquals("Invalid input", calculate_total_cost(-5, -4, 1))
14 – assertEquals("Invalid input", calculate_total_cost(0, 0, 1))
15 – assertEquals(0, calculate_total_cost(1, 4, 1))
16 – assertEquals(0, calculate_total_cost(5, 4, 1))
17 – assertEquals("Invalid input", calculate_total_cost(-5, -4, 2))
18 – assertEquals("Invalid input", calculate_total_cost(0, 0, 2))
19 – assertEquals("Invalid input", calculate_total_cost(1, 4, 2))
20 – assertEquals("Invalid input", calculate_total_cost(5, 0, 2))

نکته ای مهم درباره این برنامه این است که در حالتی که D از 0 کمتر باشد یا از 1 بیشتر باشد ،
با این که ورودی های نامعتبری است اما در این حالات خطایی به بیرون ارسال نمی شود .