

Teste de software

Programa: f1.o

Entradas possíveis:

Entradas devem ser números reais, tendo 2 entradas, x e y, sendo estes floats, todos argumentos e resultados devem ser apenas valores Reais e não devem ser números complexos

A função de cos utiliza radianos

Os códigos aceitam os valores em radianos, ou seja, um valor 2π iria ser um círculo completo

Calculo do f1 testado:

$$f1(x,y) = \text{sqrt}(x) - \cos(y)$$

Contas comuns utilizadas

$$\text{Pi}=3.141592$$

$$\cos \pi = -1$$

$$\cos 0 = 1$$

Tabela de Classes de testes

ENTRADA	CLASSE
X	$X > 0$
X	$X = 0$
X	$X < 0$
X	$X > 9999$
X	$X = \text{PI}$
Y	$Y > 0$
Y	$Y = 0$
Y	$Y > 9999$
Y	$Y < 9999$
Y	$Y = \text{PI}$
X & Y	NULL

Tabela de Testes

Teste	Valor X	Valor Y	Saída esperada	Resultado Esperado	Resultado obtido
A	4	PI	3.0	Classe válida	Ok
B	-6	PI	NAN (Not A Number)	Classe Invalida, número imaginário, resultado complexo	Falhou
C	0	1	-0.54030230586	Classe válida	Ok
D	4	-4	2.6536436209	Classe válida	Ok
E	3	PI	2.73205080757	Classe válida	Ok
F	PI	PI	2.77245385091	Classe válida	Falhou
G	0	0	-1	Classe válida	Ok
H	227	PI	16.0665191733	Classe válida	Ok
I	4.6	42.1	2.45124863585	Classe válida	Falhou
J	15.12	-25.412	2.92718440204	Classe válida	Falhou
K	∅	∅	NaN	Classe invalida, não foi possível realizar o teste	NULL
L	10 000	PI	101	Classe valida	Ok
M	4	10 000	2.95215537	Classe valida	Ok
N	4	- 10000	2. 5215536825	Classe valida	Ok

Os valores de X e Y dados em cada teste são os valores que são adquiridos após as contas, no caso raiz quadrada e cosseno, por exemplo inicialmente X valia 4, mas na explicação irá dar $X=2$ pois raiz quadrada de 4 seria 2, foi decidido mostrar o resultado de cada conta em formulas para ficar melhor a visualização dos cálculos para ter certeza que os resultados esperados estão corretos, caso o resultado esperado esteja incorreto durante os testes, uma correção será feita e notado erro feito nos testes

Na tabela de classes de testes foi removido a classe de “ $X < 9999$ ” por conta do fato que o teste B resulta em um numero imaginário, assim como daria qualquer valor abaixo de 0 quando extraído a raiz quadrada, por conta disso foi considerado que qualquer valor abaixo de 0 no X retorna um número imaginário

Teste A

$X=2$ $Y=-1$, formando $2-(-1)$

O teste A foi feito para ter certeza que em um cenário comum de 2 valores positivos funcionam, e de que o valores de PI na função cos de math.h dá o mesmo valor que o esperado

Teste B

$X = 2.44948974i$ $Y = 1$, formando $2.44948974i - (1)$

O teste B foi feito para testar valores negativos no valor X, pode ser desconsiderado pelo fato de que o valor resultante ser complexo, isso deve por conta do fato de que valores negativos à raiz quadrada dão números imaginários

O resultado dado pelo programa foi “nan” mas mesmo que dado NAN como saída esperada, o programa não entendeu que o NAN (Not A Number) seria a mesma coisa que o nan recebido como saída

```
In file included from main_testes.c:2:0:
f1_testes.h: In function 'f1_testeB':
f1_testes.h:22:45: error: 'NaN' undeclared (first use in this function); did you mean 'NAN'?
    double x = -6, y = PI, saida_esperada = NaN;
                                           ^~~~~
                                           NAN
f1_testes.h:22:45: note: each undeclared identifier is reported only once for each function it appears in
```

Teste C

$X = 0$ $Y = 0.54030230586$, formando $0 - (0.54030230586)$

O teste C foi feito para testar o valor 0 no X, para ver se caso a função dê um valor diferente na raiz quadrada de 0, o que não deve acontecer, mas testado por precaução

Primeiramente deu uma falha, mas após analisar, o resultado esperado dado retornava como falha, anteriormente dado como 0.54[...], o resultado esperado era na verdade negativo, por conta da fórmula dar o cosseno como valor negativo para subtrair da raiz quadrada de x

Teste D

$X = 2$ $Y = -0.6536436209$, formando $2 - (-0.6536436209)$

O teste D foi feito para testar valores negativos na casa do Y, caso os cálculos de cos utilizando radianos dê um resultado estranho, para ter certeza que mesmo em valores abaixo de 0 cos retorne valores dentro do esperado

Teste E

$X = 1.73205080757$ $Y = -1$, formando $1.73205080757 - (-1)$

O teste E foi feito para testar se os quando X tiver um valor que dê uma raiz quebrada, verificar caso o programa retorne o valor corretamente

Teste F

$X = 1.77245385091$ $Y = -1$, formando $1.77245385091 - (-1)$

O teste F foi feito para confirmar a reação do programa com o valor de PI, para ter certeza que mesmo que ambos valores, x e y, sejam pi, o programa continua sem erros

O teste F deu falha, não consegui encontrar nenhum erro na formula, o resultado esperado está correto e os valores de PI deveriam retornar corretamente, acredito que alguma parte do programa falha quando ambos X e Y recebem PI

O teste F falhou por conta do fato que ao chegar na 6 casa decimal após o virgula a o programa arredondou o numero para caber na saída, o qual está dada como float

Teste G

X=0 Y=1, formando 0-(-1)

O teste G foi feito para testar o que aconteceria caso ambos valores de x e y fossem 0, caso o programa se confunda com ou negue dois valores 0

Teste H

X= 15.0665191733 Y= -1, formando 15.0665191733 -(-1)

O teste H foi feito para checar se o programa aceita um número primo dentro do valor de X, também checando caso se um valor acima de 100 complica o processo

Teste I

X= 2.14476105895 Y= -0.3064875769, formando 2.14476105895- (-0.3064875769)

O teste I foi feito para verificar números decimais funcionam corretamente dentro da função, feito considerando o fato que as entradas de X e Y são dadas como float

Teste J

X= 3.88844441904 Y= 0.961260017, formando 3.88844441904- (0.961260017)

O teste J foi feito para verificar se algum erro ocorre caso Y seja decimal E negativo, o teste de X com decimal negativo não será feito por conta de qualquer negativo em X retornar um número imaginário

O teste J teve uma saída que não bate com o valor esperado, o valor esperado 2.92718440204 saiu como 2.937458, acredito que por conta da saída testada ser dada como float o teste J arredondou o 6º valor decimal para cima, da mesma forma que tinha ocorrido no teste F

Teste K

X= \emptyset Y = \emptyset , formando \emptyset - (\emptyset)

O teste K foi feito para verificar a reação do programa com entradas nulas, esperado retornar erro ou apenas não retornar nada

O teste K foi cancelado e substituído por conta de não ser possível dar valores vazios para os testes

Teste L

$X = 100$ $Y = -1$, formando $100 - (-1)$

O teste J foi feito para verificar como o sistema reage com valores limites, no caso testando o 10 000 dentro do valor X

Teste M

$X = 2$ $Y = -0.95215537$, formando $2 - (-0.95215537)$

O teste M foi feito com o objetivo de repetir o teste L, mas neste caso com o 10 000 no valor de Y

Teste N

$X = 2$ $Y = -0.95215536825$, formando $2 - (-0.95215536825)$

O teste N foi feito para verificar a reação do sistema com um numero limite negativo no Y, como números negativos em X resultam em números imaginários e consequentemente, retorna números complexos, o teste foi limitado apenas para o Y

Conclusão:

Em todos testes que retornaram apontam que dentro desse programa existem 2 casos de retornos incorretos que foram encontrados:

- 1- Caso a saída do programa receba um valor com números após a 6ª casa decimal
- 2- Caso o X receba um número negativo

Caso contrário o programa aparenta funcionar de maneira nominal, resultando os mesmos valores que os esperados, isso se deve por conta da simplicidade do cálculo, apenas em poucos casos a saída do programa retornaram um valor inesperado

O programa em si não aparenta apresentar falhas, foi encontrado apenas alguns erros de compatibilidade com o programa de teste e as saídas recebidas do programa em questão, o arredondamento automático que ocorre quando a saída dada do programa testado entra no programa de testes foi apenas descoberto após ser feito um ajuste no programa de teste para verificar as saídas que receberam um “Falhou!”