



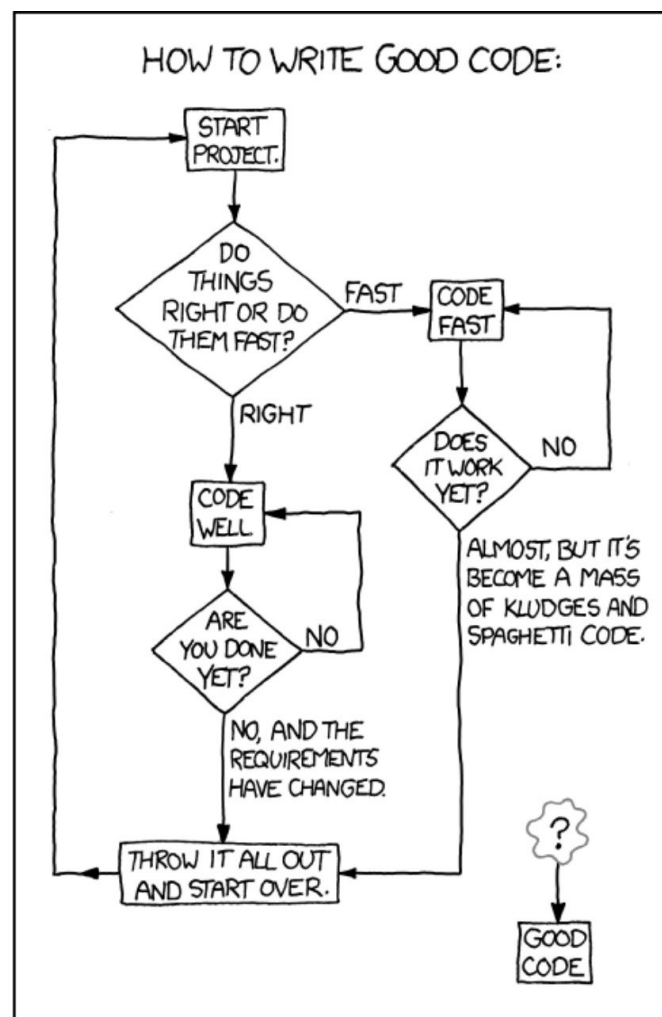
OPDRACHT 4

FLOOR VAN DE LEUR

GEBASEERD OP EEN EERDERE OPDRACHT VAN:

JOSÉ LAGERBERG - ROBIN DE VRIES - JETSKIE BEKS - MARCO BROHET - INGRID KOKKEN

Dr. Quackerjack



Deadline: vrijdag 30 september 2022, 23:59

1 Introductie

In de afgelopen vier weken heb je de basis geleerd van programmeren in C. Bij de voorgaande opdrachten moest je zelf een oplossing voor een probleem verzinnen en implementeren. Daarbij hebben we het belang benadrukt van een nette, consistente programmeerstijl en van commentaar dat je code verduidelijkt.

In deze opdracht zul je goed merken waarom een nette stijl en duidelijk commentaar belangrijk zijn. Je krijgt dit keer de code al aangeleverd, dus die hoeft je niet zelf te schrijven. Er zijn wel een paar problemen: het programma compileert niet, een deel van de functionaliteit is onjuist geïmplementeerd, en het gebrek aan stijl en commentaar maakt het debuggen van het programma er niet gemakkelijker op.

Een belangrijk onderdeel van deze opdracht is het correct implementeren van twee algoritmes. Je mag de werking van de benodigde algoritmes opzoeken, maar je moet in je commentaar wel in je eigen woorden uitleggen hoe de algoritmes werken. Vermeld de bronnen die je gebruikt hebt ook in je commentaar. Let op: je mag wel pseudocode gebruiken die je online hebt gevonden, maar je mag geen 'echte' code van internet gebruiken!

Tijdens het nakijken behandelen we je ingeleverde programma alsof het je eigen code is. Dat betekent dat je over iedere keuze op het gebied van implementatie, structuur, design, stijl en commentaar moet nadenken – je komt niet weg met “maar het stond zo in de aangeleverde code”.

2 De casus

Dr. Quackerjack – een moleculair bioloog – is bezig met een onderzoek naar het zeer gevaarlijke virus COVID-21, dat verspreid lijkt te worden door eenden. Om dit te bewerkstelligen heeft hij met zijn team DNA-monsters afgenomen bij dode en levende eenden. Het DNA van deze monsters is gesequenced en de relevante stukken zijn opgeslagen in korte strings (bijvoorbeeld ATCGCTGTA). Hierna heeft dr. Quackerjack geprobeerd een C-programma te schrijven om het DNA te analyseren en relaties te kunnen vinden tussen de opgeslagen strings, met als doel het ontwikkelen van betere detectiemethoden voor COVID-21.

Omdat dr. Quackerjack geen ervaren C-programmeur is en niet zo stressbestendig is, werkt het programma helemaal niet en is de code grotendeels onleesbaar. Aan jou de taak om het programma te herstellen en te verbeteren. Zorg ervoor dat alle functionaliteit naar behoren werkt, dat er voldoende commentaar aanwezig is (met een uitleg van de gebruikte algoritmes) en dat de stijl netjes en consistent is.

Dr. Quackerjack en zijn team hebben helaas geen tijd om persoonlijk te corresponderen over de code. Ze hopen dan ook dat je zelf uit kunt *vogelen* wat er gedaan moet worden. En het heeft haast, want op dit moment vliegt er een enorme zwerm geïnfecteerde eenden richting Nederland, die naar verwachting in de nacht van zaterdag op zondag onze grenzen oversteekt. Lukt het jou om voor de deadline dr. Quackerjacks programma af te maken en zo Nederland te redden?

3 De opdracht

Begin met het leesbaar maken van het programma, zodat de structuur een beetje duidelijk wordt. Daarna kun je het programma proberen te compileren. Werk van boven naar beneden door de errors en warnings heen en los ze op.

Als alle foutmeldingen weg zijn, kun je kijken naar de algoritmes en de rest van de functionaliteit. Dr. Quackerjack had veel plannen voor dit programma, maar hij heeft ze niet allemaal geïmplementeerd. De voorbeelduitvoer laat zien wat jouw versie van dr. Quackerjacks programma moet kunnen. Vergeet ook niet om goed commentaar bij de functies en algoritmes te zetten. Zo kan dr. Quackerjack nog wat van zijn fouten leren.

Let op: de uitvoer van je programma moet exact overeenkomen met de voorbeelduitvoer.

3.1 Inleveren

Lever opdracht4.c in op Canvas. De Makefile hoeft je niet in te leveren. CodeGrade gebruikt de Makefile die ook in het aangeleverde framework zit, dus zorg dat je programma daarmee correct werkt.

4 Voorbeelduitvoer

Onderstaande voorbeelduitvoer laat zien wat je programma allemaal moet kunnen en hoe het moet reageren op verschillende situaties. Zorg dat je de voorbeelduitvoer exact aanhoudt: iedere letter en spatie moet kloppen (de regels in de *list of commands* beginnen met twee spaties, en de ruimte tussen “compare” en “compare two strings” is ook twee spaties).

```
$ ./opdracht4
Welcome to DNA Matcher v0.2
console> help
LIST OF COMMANDS
  add ...          add to database
  compare ... ..   compare two strings
  help            print this text
  list            print database
  read ...         read from file and add to database
  remove ...       remove from database
  retrieve ...     find in database
  size            print number of items in database
  quit            stop
console> size
0 items in database
console> list
Database is empty
console> add acgt
"acgt" added to database
console> add agtc
"agtc" added to database
console> add CATG
"catg" added to database
console> size
3 items in database
console> list
acgt
agtc
catg
console> add acgt
"acgt" not added; already in database
console> remove acgt
"acgt" removed from database
console> list
agtc
catg
console> remove acgt
"acgt" not removed; not in database
console> remove agtc
"agtc" removed from database
console> remove catg
"catg" removed from database
console> retrieve cat
No match found; database is empty
console> read input.txt
"dog" added to database
"horse" added to database
"cat" added to database
"frog" added to database
console> list
dog
horse
cat
frog
```

```

console> retrieve cat
Perfect match found; "cat" is in database
console> retrieve duck
No perfect match found; "duck" is not in database
Best matches:
Distance      String
3             dog
4             cat
4             frog
console> read test.txt
"test.txt" not found
console> compare quacker jack
0 1 2 3 4
1 1 2 3 4
2 2 2 3 4
3 3 2 3 4
4 4 3 2 3
5 5 4 3 2
6 6 5 4 3
7 7 6 5 4
Difference = 4
console> compare cat BAT
0 1 2 3
1 1 2 3
2 2 1 2
3 3 2 1
Difference = 1
console> add
Please enter a valid command
console> add abc def
Please enter a valid command
console> cookie
Please enter a valid command

...

console> add gtca
"gtca" not added; database is full
console> quit

```