Informatica: C# WPO 7

1. Inhoud

1D-arrays, Lijsten

2. Oefeningen

- Demo 1: Vul de 1D-array
- Demo 2: Stringreplace
- Demo 3: Vul de lijst
- Demo 4: Debug oplossingen demo's 1, 2 en 3
- A: Array reversal
- A: Gemiddelde
- A: Zoek het aantal voorkomens
- A: Min-max
- A: Interpoleer
- E: Zeef van Eratosthenes
- E: Bubble Sort
- X: Running average

2.1 Demo 1: Vul de 1D-array

Schrijf een programma waarin een 1D-array gebruikt wordt. Deze array wordt oplopend gevuld door gebruik te maken van een loop. Nadien worden de waarden van deze array geprint binnen een tekstbox door gebruik te maken van een andere loop.

2.2 Demo 2: Stringreplace

Schrijf een applicatie die de gebruiker toelaat om in een stukje tekst bepaalde karakters te vervangen door andere. Het aantal verschillende karakters dat je laat veranderen is beperkt tot 5. Deze mogen vast in het programma ingegeven worden.

2.3 Demo 3: Vul de lijst

In deze demo wordt het gebruik van een lijst gedemonstreerd. Hierbij wordt de eerste demo hernomen. De verschillen tussen een lijst en een array worden hier extra benadrukt.

2.4 Demo 4: Debug oplossingen demo's 1, 2 en 3

Ook hier wordt de debugger gebruikt om het verloop van de programma's in de voorgaande demo's te illustreren.

2.5 A: Array reversal

Laat de gebruiker toe om 10 getallen in te geven. Gebruik eerst een array met 10 plaatsen om de waarden bij te houden. Eenmaal de 10 getallen ingegeven zijn, worden alle getallen eerst in volgorde afgeprint in een label. Eens de getallen afgeprint zijn, worden de elementen binnen de array omgewisseld. Print daarna de array opnieuw af in een andere label. Hieronder worden 2 voorwaarden opgelegd.

1. Wissel de waarden van de array om en sla ze op in een nieuwe array.
2. Wissel de waarden van de array om en sla ze in dezelfde array op zonder gebruik te maken van een nieuwe array.

2.6 A: Gemiddelde

Schrijf een programma dat 10 elementen inleest en deze in een array opslaat. Van zodra er 10 elementen ingelezen zijn, wordt het gemiddelde van deze 10 elementen bepaald. In de eerste versie schrijf je dit door gebruik te maken van een gewone array. In een tweede versie gebruik je hiervoor een lijst. Beide oplossingen mogen in hetzelfde programma geschreven worden.

2.7 A: Zoek het aantal voorkomens

Laat de gebruiker toe om een string via een tekstbox in te geven. Maak een variabele aan (char) waarin je een teken in bewaart. De string kan je beschouwen als een array van char. Ga nu alle tekens binnen de string af en tel het aantal keren dat dit teken hierin voorkomt.

2.8 A: Min-Max

Schrijf een programma waarin een array met 10 getallen opgenomen wordt. De getallen worden door de gebruiker via een button en textbox ingegeven. Nadat 10 getallen zijn ingegeven worden de grootste en kleinste waarde uit de array gezocht. Print zowel het grootste als het kleinste getal af.

2.9 A: Interpoleer

In onderstaande tabel worden alle getallen gegeven die op een even index staan. Het is nu aan jou om de waarden op de oneven plaatsen te berekenen. Die zijn momenteel aangeduid met "X". Dit doe je door het gemiddelde te nemen van de naburige elementen. Print de hele array af in een tekstbox. Op hoeveel elementen moet je de array voorzien?

Index	Waarde
0	10
1	X
2	15
3	X
4	18
5	X
6	10
7	X
8	-10
9	X
10	-50

2.10 E: Zeef van Eratosthenes

De zeef van Eratosthenes is een gekende methode om priemgetallen te vinden. Het algoritme werkt als volgt:

- maak een array aan met een lengte van 100 van het type bool,
- zet alle elementen in de array op true,
- start bij getal 2,
- alle veelvouden van 2 worden op false gezet (2 niet),
- ga naar getal 3 en zet daarvan alle veelvouden op false,
- getal 4 staat op false, dus deze wordt overgeslagen,
- 5 staat op true. Alle veelvouden van 5 worden op false gezet,
- ga zo door totdat getal 100 bereikt wordt.

Nadat alle getallen doorzocht zijn, kunnen de priemgetallen geprint worden. Itereer over de volledige array. Indien het huidige getal op true staat, wordt dit getal geprint, anders niet (dit is een priemgetal). Als alles goed verloopt, worden enkel de priemgetallen afgeprint. Hieronder wordt een fragment met de inhoud van de array weergegeven (true betekent priemgetal).

Index	1	2	3	4	5	6	7	8	9	10
Waarde	true	true	true	false	true	false	true	false	false	false

2.11 E: Bubble Sort

Bubble Sort is een eenvoudig algoritme dat toelaat om een rij waarden van klein naar groot te sorteren. Het algoritme werkt als volgt:

1. Start bij het begin van de rij.
2. Vergelijk de 1^{ste} waarde met de 2^{de}. Indien de 2^{de} waarde kleiner is dan de 1^{ste}, worden beide waarden gewisseld binnen de array.
3. Vergelijk nu de 2^{de} met de 3^{de} waarde en voer dezelfde wisseloperatie uit indien nodig.
4. Ga zo door totdat alle elementen van de array afgegaan zijn.
5. Start opnieuw bij het begin (punt 2) en herhaal dit totdat alle elementen zijn afgegaan.

2.12 X: Running average

Het running average principe wordt veel toegepast in het o.a. opschonen van ruwe gemeten signalen. I.p.v. het gemiddelde te berekenen van het hele signaal, wordt er enkel een fractie uitgemiddeld. Dit lokaal uitmiddelen loopt over het hele signaal heen, wat de naam van running average rechtvaardigt. In formulevorm wordt het running average beschreven als:

$$RA_n = \frac{1}{N} \sum_{k=0}^{N-1} x_{n-k} \tag{1}$$

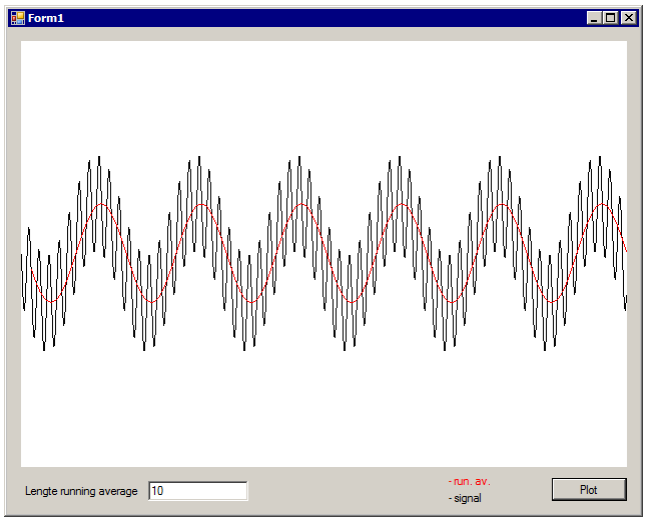
waarbij RA_n het berekende gemiddelde is op plaats n, N het aantal punten waarover het gemiddelde effect heeft, en n het element. Dit principe wordt hieronder met een getallenreeks verduidelijkt:

Waarde	5	4	6	8	1	4	0	1	2	5	6	9	5	8	9	6	5	1	3	6	0
RA	0	0	0	6	5	5	4	2	2	2	4	6	6	7	8	7	7	5	4	4	3

Tabel 1: Voorbeeld van een running average, de lengte van de running average is hier op 4 elementen gekozen. Vandaar dat de eerste 3 resultaten een nul-waarde bevatten.

Schrijf een programma dat twee sinussen met elkaar optelt en plot deze functie. Voorzie een textbox waarin de lengte van de running average kan worden ingegeven. Pas het running average principe toe op deze functie en plot dit over het signaal. Indien geen lengte voor de running average wordt opgegeven wordt deze niet berekend en ook niet getekend. Een voorbeeld is weergegeven in figuur 1. De sinussen die met elkaar opgeteld moeten worden zijn:

$$\begin{aligned} y(t) &= 50 \cdot sin(2 \cdot \pi \cdot 0.01 \cdot t) \\ y(t) &= 50 \cdot sin(2 \cdot \pi \cdot 0.1 \cdot t) \end{aligned} \tag{2}$$



Figuur 1: Voorbeeld van een running average (zwart = origineel signaal, rood = running average).