

# Doel van de oefeningen

Deze oefeningen zijn oefeningen om je programmeervaardigheden te helpen ontwikkelen. Je maakt deze oefeningen zelf op je eigen ritme. Voor alle vragen omtrent de oplossingen raadpleeg je best je docent tijdens de les, zeker als je twijfelt of je oplossing correct is (ook al werkt je programma, daarom is de werkwijze nog niet altijd correct).

## Organisatie van je oefeningen

Je kan meerdere programma's schrijven in 1 visual studio Solution.

- Voor elke oefening voeg je dan aan je solution een nieuw project toe (Visual C# → windows application → console application)
- Kies een duidelijke naam bvb oefening1\_1
- Om dit nieuwe project actief te maken (zorgen dat deze applicatie runt als je op debug drukt) klik je in je solution explorer met de rechter muisknop op je project en selecteer je "Set as StartUp project. (dit hoeft natuurlijk niet voor je eerste project)

## Tips

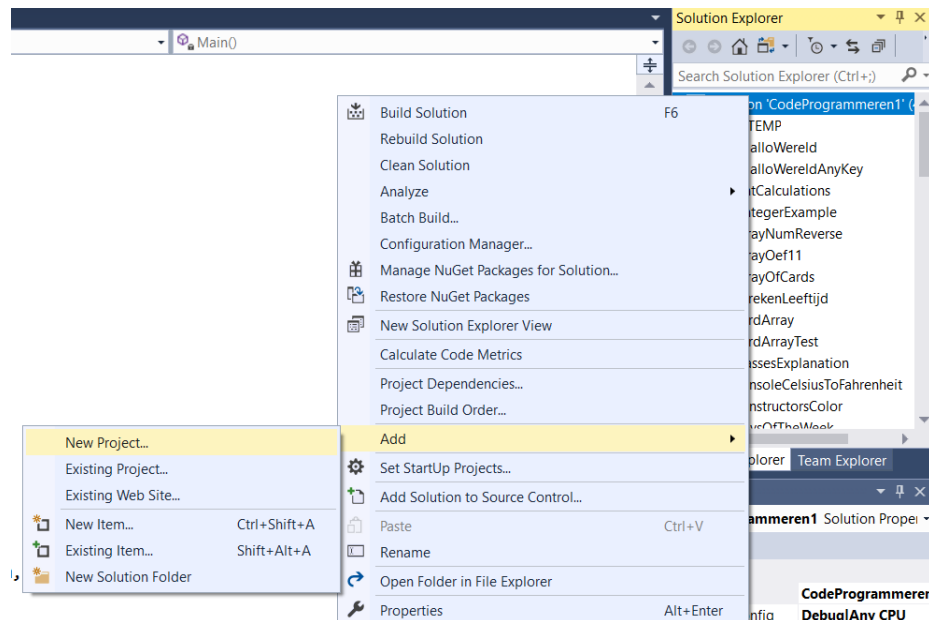
- Bekijk eerst de theorie vooraleer je start, en zorg dat je alles goed begrijpt.
- Is er toch iets niet 100% duidelijk of twijfel je, vraag dan eerst uitleg aan je docent.
- Is de docent even niet beschikbaar, kan je altijd al hulp opzoeken in de msdn.
- Probeer steeds je oefeningen te maken zonder telkens naar voorbeelden te kijken, maar werk vanuit je kennis.
- Staan er fouten in je code zoek dan je foutmelding op, zoek hulp in de msdn, ga er niet van uit dat alle informatie op het internet betrouwbaar is.
- Debuggen is zeker zo belangrijk als programmeren, loopt er iets mis in je programma, is debuggen en door je code stappen de eerste stap.
- Maak de oefeningen zeker ZONDER kopiëren/plakken uit voorbeelden/msdn. Dit zorgt er enkel voor dat je elementen overbrengt, maar is geen goede leermethode. Probeer dit ten alle tijden te vermijden.
- Als je merkt dat je nog naar voorbeelden/theorie/msdn moet teruggrijpen, is het een teken dat je nog niet alles volledig onder de knie hebt, blijf oefenen (hermaak oefeningen, of variaties daarop). Ook bij het hermaken van oefeningen, kijk niet naar vorige oplossingen, maar werk vanuit je kennis.
- Wil je bijkomende oefeningen:
  - Bedenk zelf variaties op deze oefeningen, evt samen met anderen in de klas zodanig dat je ook samen de oplossing kan bekijken
  - Zoek oefeningen online op

- Ook hierover mag je altijd vragen stellen

# Oefeningen: introductie tot basisapplicaties & Variabelen

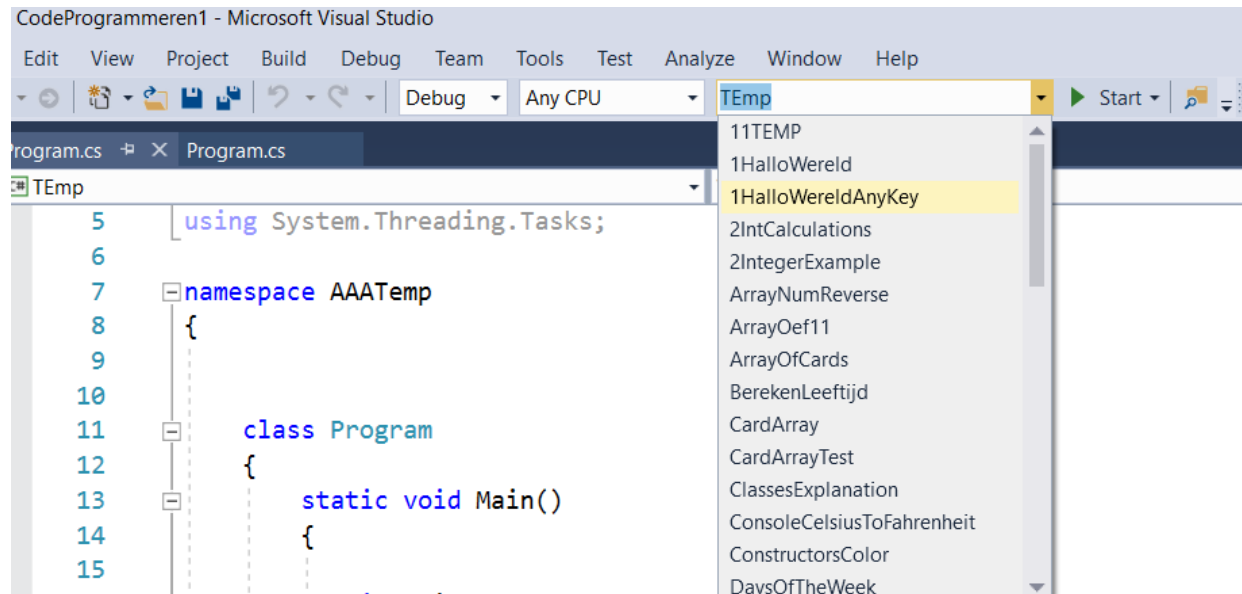
## Basisoefeningen

1. Maak een nieuw project aan en schrijf een programma dat “Hello World” uitschrijft, maar in 2 aparte instructies (dus eerst “Hello” en daarna “World”). Zorg ook dat je de gebruiker vraagt om met een druk op de toets het programma af te sluiten en vang dit op. Let op de spatie tussen “Hello” en “World”. Kan je verschillende manieren bedenken om die spatie erin te krijgen? Probeer uit!
2. Werk met meerdere projecten in Visual studio, binnen 1 solution. In je solution explorer:
  - a. Klik met je rechtermuis op je solution
  - b. Ga naar Add
  - c. Selecteer naar New Project

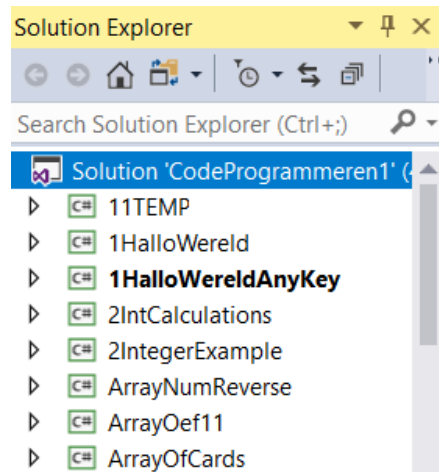


- d. Maak een nieuw project aan (met als naam 1HelloWorldAnyKey)

- e. Vergeet niet je project als startup project in te stellen, anders voer je een andere applicatie uit uit een vorig project. Dit selecteer je bovenaan



- f. Je ziet het ook in de solution explorer, dit project staat in het vet (alternatieve methode is dat je ,met rechtermuis op je project klikt en Set As Startup Project selecteerd)



Herschrijf nu je vorige oefening, maar zodanig dat “Hello” en “World” op aparte regels worden afgedrukt (zonder spatie dit keer). Zorg er nu voor dat “World” pas wordt afgedrukt nadat de gebruiker op ENTER heeft gedrukt. Sluit ook het programma pas af nadat “World” getoond werd aan de gebruiker en deze opnieuw op ENTER drukt.

3. Debug je vorige applicatie door stap per stap door de code te steppen, dit doe je door eerst een breakpoint te plaatsen op je eerste regel code die je zelf hebt toegevoegd.

Daarna kan je telkens een regel verder “steppen”. Doorloop zo heel je programma tot het afloopt. Gebruik `step into` om door de code te gaan.

4. Herschrijf nu je vorige oefening, maar zodanig dat “Hello” en “World” op aparte regels worden afgedrukt maar met 1 Writeline instructie. Maak gebruik van een escape sequence (zoek escape sequences op in MSDN).  
Sluit ook het programma pas af nadat “World” getoond werd aan de gebruiker en deze opnieuw op ENTER drukt.
5. Schrijf een programma dat de voornaam van de gebruiker inleest en vervolgens de gebruiker groet met 'Hello, *first\_name*!'.
6. Schrijf een programma dat volgende tekst uitschrijft:  
    “Hello, my name is:   XXX”  
waarin je XXX vervangt door je eigen naam en voornaam, voor je naam staan geen spaties maar een tab (gebruik een escape sequence)
7. Herschrijf je vorige programma zodanig dat elk woord door een aparte instructie wordt uitgeschreven.
8. Zoek in de msdn op hoeveel versies er zijn van `Console.Write`
9. Zoek in de msdn op welke speciale karakters de `Console.WriteLine` extra toevoegt t.o.v. de `Console.Write`. Kan je deze zelf toevoegen aan een `Console.Write(“Hello World”)` zodat die toch werkt zoals een `Console.WriteLine(“Hello World”);`

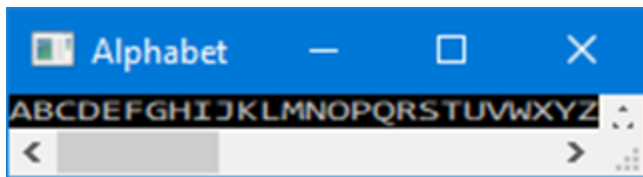
## Extra

Nieuwe commando's voor de volgende oefeningen, zoek de uitleg op in de MSDN:

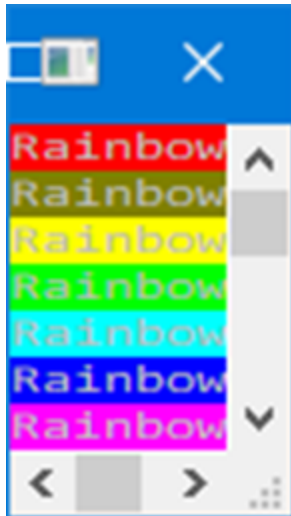
- **`Console.SetWindowSize(100, 50);`**  
Verandert de grootte van het scherm
- **`Console.Title = “Test ABC”;`**  
Verandert de titel bovenaan het scherm naar “Test ABC”
- **`Console.ForegroundColor = ConsoleColor.Red;`**  
Verandert de kleur van de tekst naar rood
- **`Console.BackgroundColor = ConsoleColor.DarkYellow;`**  
Verandert de achtergrond van de tekst naar donkergeel

Probeer volgende schermen na te maken met bovenstaande commando's:

1.



2.



3.



4. Dit filmpje:

<https://i.imgur.com/kprQGag.gifv>

# Oefeningen: Bewerkingen

## Basisoefeningen





1. Ontwerp een toepassing waarmee je een ingevoerd bedrag in euro's kan omrekenen naar dollar. Gebruik hierbij de volgende wisselkoers:  $1000 \text{ EUR} = 1173\$$ . Werk met gehele getallen. Het resultaat in de console wordt als volgt weergegeven: "X EUR is gelijk aan Y USD".
2. Schrijf een programma dat twee gehele getallen initialiseert op 3 en 8. Tel de twee getallen bij elkaar op en sla het resultaat op in een nieuwe variabele. Schrijf het resultaat naar het scherm.
3. Schrijf een programma dat twee getallen initialiseert op 3.3 en 8.8. Tel de twee getallen bij elkaar op en sla het resultaat op in een nieuwe variabele. Schrijf het resultaat naar het scherm.
4. Schrijf een programma dat de gebruiker vraagt om vijf getallen en schrijf daarna het gemiddelde van die vijf getallen naar het scherm (dit mag afgerond).
5. Ontwerp een toepassing waarmee je een de temperatuur in graden Celsius kan omrekenen naar Fahrenheit. De formule die je hiervoor kan gebruiken is:  
 $F = (9/5) \times C + 32$ . Het resultaat in de console wordt als volgt weergegeven: "A Temperature of x degrees Celsius corresponds to y degrees Fahrenheit."
6. In de KdG Winkel zijn de prijzen van alle artikelen weergegeven exclusief BTW. Ontwerp een toepassing waarmee je met de opgegeven prijs exclusief BTW de prijs inclusief BTW kan berekenen. Het BTW-tarief is 21 %. Het resultaat in de console wordt als volgt weergegeven: "De prijs inclusief BTW bedraagt: x"
7. De plaatselijke voetbalclub organiseert een mossselfestijn. Naast mosselen met frietjes (20 EUR) bieden ze voor de kinderen de mogelijkheid om een koninginnehapje (10 EUR) te kiezen. Verder is er een ijsje als nagerecht voorzien (3 EUR). Om het gemakkelijk te maken kosten alle dranken 2 EUR. Ontwerp een consoletoepassing zodat de vrijwilliger aan de kassa alleen maar de juiste aantallen moet ingeven om de

totaal te betalen prijs te berekenen. Het resultaat in de console wordt als volgt weergegeven: "Het totaal te betalen bedrag is x EUR".

8. De inkomprijs voor KdGPretLand bedraagt 30 EUR voor volwassenen en 15 EUR voor kinderen tot 12 jaar. Ontwerp een consoletoepassing waarmee de kassier het aantal kinderen en volwassen kan ingeven om zo het totaal te betalen bedrag te berekenen. Voorzie een overzichtelijke weergave in de console.
9. Schrijf een programma dat de gebruiker vraagt om het huidige jaartal en om het jaar waarin hij geboren is. Reken uit hoe oud de gebruiker is aan het eind van het huidige jaar (in hele jaren) en schrijf het antwoord naar het scherm.
10. Schrijf een programma waarmee je het laatste woord uit een naam kan halen. Vraag aan de gebruiker een volledige naam in te geven. Vraag vervolgens aan de gebruiker de lengte van het laatste woord in die naam. Werk daarna iets uit dat het laatste woord ophaalt en afbeeldt. Bijv: uit "David Van Hoecke" wordt het woord "Hoecke" gehaald.
11. Bereken hoeveel uren vrije tijd je in een week hebt. Vraag aan de gebruiker hoeveel uur les hij heeft, en bereken aan de hand daarvan hoeveel uur hij die week vrij heeft om andere dingen te doen.  
Hou er rekening mee dat je voor elke 2 uur les, 1 uur thuis studeert/werkt.  
Hou er ook rekening mee dat je ook nog elke dag moet slapen! Je slaaptijd telt niet als vrije tijd.  
Druk de resterende vrije tijd af.
12. Schrijf een programma dat een aantal seconden vraagt (bv 4000) en dan het aantal uren, minuten en seconden berekent die hierin passen.  
  
Voorbeelden:
  - 70 seconden wordt 1 minuut en 10 seconden
  - 130 seconden wordt 2 minuten en 10 seconden
  - 3601 seconden wordt 1 uur, 0 minuten en 1 seconde
  - 4000 seconden wordt 1 uur, 6 minuten en 40 seconden zijn.
13. Kies minstens 2 van je voorgaande oefeningen uit en step doorheen heel het programma. Bekijk de variabelen en hun waarden eens vanuit de debug vensters: Local, Autos en Watch.
14. Schrijf een programma dat  $2^{125}$  berekent, of **2x2x2x...x2 125** keer. Maak slim gebruik van variabelen om geen 125 keer  $2*2$  te moeten schrijven (let op dit getal past niet in een integer).

15. Schrijf een programma dat de wortel berekent van een door de gebruiker ingevoerd getal.
16. Schrijf een programma dat **de derdemachtswortel** van een ingegeven getal berekent.
17. Schrijf een programma dat de wortel berekent van een ingegeven getal, afgerond tot 2 cijfers na de komma.
18. Schrijf een dronken papegaai applicatie: je voert tekst in. De papegaai herhaalt wat je hebt ingegeven. Daarna herhaalt die alles al roepend (alles in hoofdletters). Daarna herhaalt die alles zonder klinkers. Daarna zegt die tussen elk woord van je originele zin "I want a cracker !!!". Zo zal "Dag papegaai, hoe gaat het er mee?" resulteren in  
"Dag papegaai, hoe gaat het er mee?"  
"DAG PAPEGAAI, HOE GAAT HET ER MEE?"  
"Dg ppg, h gt ht r m?"  
"Dag I want a cracker !!! papegaai, I want a cracker !!! hoe I want a cracker !!! gaat I want a cracker !!! het I want a cracker !!! er I want a cracker !!! mee?"

## Extra Oefeningen

19. Schrijf een programma dat vraagt om je naam en voornaam (in 1 ReadLine instructie) en die omgekeerd in beeld brengt. Zo zal "Joop Peeters" resulteren in "Peeters Joop".
20. Deel een van je programma's met een klasgenoot en laat die je programma runnen. M.a.w. Zorg ervoor dat de juiste exe file bij een klasgenoot geraakt.
21. Herschrijf je dronken papegaai applicatie zodat die antwoordt met speciale tekens zoals           .
22. Laat een gebruiker 2 getallen ingeven. Toon true in beeld als het eerste getal kleiner is dan het tweede.
23. Schrijf een oefening die een tijd inleest als u:min:seconden (als 1 string). En herbereken dit naar enkel seconden (dus reken de uren en minuten om naar seconden, en bereken het totaal aantal seconden). Gebruik substring en IndexOf om de ':' posities te vinden



# Oefeningen: Typecasten

## Basisoefeningen

1. Schrijf een programma dat de gebruiker vraagt om drie gehele getallen en schrijf daarna het gemiddelde van die drie getallen naar het scherm (niet afronden, maar als kommagetal).
2. Maak zelf een ToUpper programma zonder de methode ToUpper van string te gebruiken. Ga ervan uit dat gebruikers 1 klein character ingeeft. Je applicatie maakt er een hoofdletter van. Gebruik de ASCII tabel.
3. Schrijf een programma dat een kommagetal inleest en dit uitschrijft
  - a. Als een geheel getal, maar afgekapt na de komma
  - b. Als een afgerond geheel getal
4. Schrijf een programma dat een unicode waarde (een getal) inleest en het symbool waarmee het overeenkomt in de unicode of ascii tabel uitschrijft.
5. Schrijf een programma dat een schrijftteken (unicode) inleest en het getal waarmee het overeenkomt in de unicode of ascii tabel uitschrijft (gebruik een impliciete conversie).
6. Schrijf een programma dat m.b.v. roman cypher 3 leestekens verschuift in het alfabet. Je vraagt aan de gebruiker hoeveel de letters in het alfabet verschoven moeten worden, bvb bij getal 3 wordt een 'a' een 'd' en 'b' een 'e', let op, op het einde van het alfabet moeten de letters ook terug naar het begin gaan, bvb bij 3 wordt een 'z' een 'c' en 'y' een 'b'. Ga uit van een ingevoerde tekst van 4 letters. Later zien we lussen en arrays om een meer elegante oplossing te bekomen.
7. Breidt vorige oefening uit zodat je ook terug vertaald naar de originele tekst, dus versleutelde tekst en sleutel hem terug naar het origineel
8. Schrijf een programma dat je het aantal letters uit een string laat zien.

# Oefeningen: Conditie / Selecties / If

## Basisoefeningen

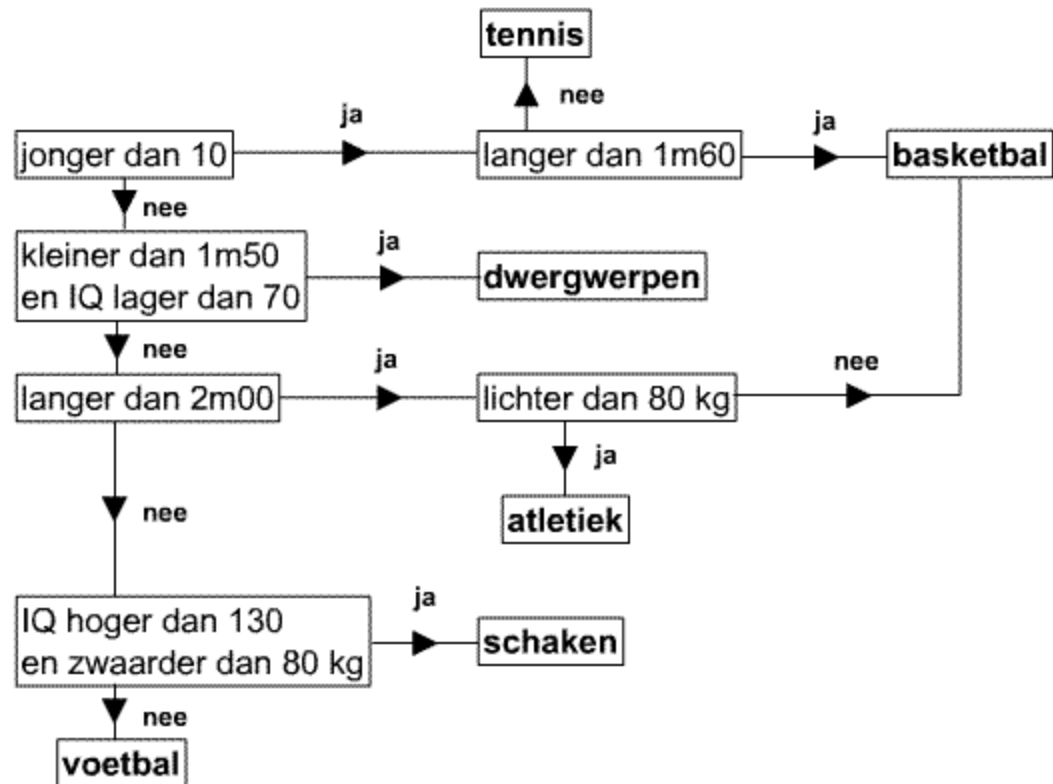
1. Om deel te nemen aan de spelen van de verkiezingen moet je minimum 18 jaar oud zijn. Ontwerp een consoletoepassing waarmee je het geboortjaar van de gebruiker opvraagt. Als de ingevoerde leeftijd minimum 18 jaar bedraagt, krijgt de gebruiker toegang tot de stemcomputer en verschijnt de tekst: "Je bent x jaar oud. Je mag stemmen". Indien niet voldaan is aan deze voorwaarde verschijnt de tekst: "Om deel te nemen aan de verkiezingen moet je minimum 18 jaar oud zijn.". Als het geboortjaar na het huidige jaar ligt, schrijf dan naar het scherm 'Je bent nog niet geboren.'.
2. Pas de vorige oefening aan zodat dit werkt op de gehele datum, dus ook de dag en de maand.
3. Laat een gebruiker 2 namen ingeven. Toon true in beeld als de eerste naam alfabetisch voor de tweede komt (let op met hoofdletters).
4. Ontwerp een consoletoepassing waarmee de uitbater van een schoenwinkel het totaal te betalen bedrag kan berekenen. Als je 2 of meer paar schoenen koopt, dan krijg je 25% korting. Een paar schoenen kost 100 EUR. Zorg ervoor dat als je een negatief getal of ongeldig getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
5. De solden lopen op hun einde en de uitbater van de Fnac geeft speciale kortingen. Ontwerp een consoletoepassing waarmee de uitbater van de Fnac het totaal te betalen bedrag kan berekenen. Als je één paar schoenen koopt, krijg je 20 % korting, bij de aankoop van 2 paar schoenen krijg je 40% korting en bij aankoop van meer dan twee paar schoenen zelfs 60 % korting. Een paar schoen kost 125 EUR. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd. En controleer ook op geheel ongeldige invoer (geen getal).
6. Ontwerp een consoletoepassing waarmee je aan de gebruiker het resultaat van 5 examens opvraagt. De opgevraagde resultaten zijn de behaalde punten op 100. De student is geslaagd als hij hoogstens 1 onvoldoende heeft en een gemiddelde van minstens 50 % behaalde. Schrijf ook het behaalde percentage uit. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
7. Schrijf een programma dat een jaartal inleest en uitschrijft of het een schrikkeljaar is. Een jaar is een schrikkeljaar wanneer het deelbaar is door 4, maar niet door 100, het is echter toch een schrikkeljaar wanneer het deelbaar is door 400.
8. Ontwerp een consoletoepassing waarmee je het lidgeld voor de Zoo berekent. Het lidgeld is afhankelijk van de leeftijd. Het lidgeld bedraagt 5 EUR voor iedereen jonger

dan 26 jaar, anderen betalen 10 EUR. Zorg ervoor dat als je een negatief of ongeldig getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.

9. Maak een applicatie die 2 namen vraagt en controleert welke naam alfabetisch eerst komt. Zorg ervoor dat deze applicatie ook werkt als een van de 2 namen in hoofdletters is.
10. Ontwerp een console toepassing waarmee je berekent hoeveel iemand voor een bioscoopkaartje moet betalen. De volle prijs van een bioscoopticket bedraagt 13,70 EUR. Op basis van de leeftijd wordt de prijs bepaald:
  - Jonger dan 5 jaar: gratis;
  - Tussen de 5 en de 12: halve prijs;
  - Tussen de 13 en de 55: vol tarief;
  - 55+: gratis
11. Schrijf een applicatie die vraagt of je de omtrek (C voor circumference) of oppervlakte (S voor surface) van een cirkel wenst te berekenen, vraag daarna naar de straal van de cirkel. Gebruik voor de berekening de waarde van PI uit de Math klasse. Zorg ervoor dat als je een negatieve straal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd. Doe de geldige invoercheck met 1 selectie
12. Pas opgave 1 aan zodat het effectief werkt op de huidige datum (gebruik `DateTime.Now`)

## Extra Oefeningen

13. Schrijf opnieuw een programma dat m.b.v. roman cypher 3 leestekens verschuift in het alfabet. Je vraagt aan de gebruiker hoeveel de letters in het alfabet verschoven moeten worden, bvb bij getal 3 wordt een 'a' een 'd' en 'b' een 'e', let op, op het einde van het alfabet moeten de letters ook terug naar het begin gaan, bvb bij 3 wordt een 'z' een 'c' en 'y' een 'b'. Nu kan je ook een if gebruiken ipv %.
14. Breidt vorige oefening uit zodat je ook terug vertaald naar de originele tekst, dus versleutel de tekst en sleutel hem terug naar het origineel.
15. Breidt vorige oefening uit zodanig dat je zowel positieve als negatieve verschuivingen kan hebben (+3 of -5,...) zodat je met 1 programma in 2 richtingen kan vertalen. Vraag de verschuiving aan de gebruiker.
16. Zet het onderstaande schema om in C#. Zorg dat je alles aan de gebruiker vraagt, maar enkel indien nodig. Reken in cm en leeftijd in jaren het programma geeft terug welke sport daarbij past.



24. Vraag de gebruiker om een naam ingeven, toon in het scherm

- De lengte van de ingevoerde naam
- De naam omgevormd naar allemaal hoofdletters
- De naam omgevormd naar allemaal kleine letters
- Als de lengte even is geef je de eerste 3 tekens van de naam, anders de laatste 3. Let op met namen korter dan 3 tekens.

## Oefeningen: Switch

### Basisoefeningen

- Schrijf een programma dat de gebruiker vraagt of hij wil optellen, aftrekken, vermenigvuldigen, delen of modulo rekenen. Vraag daarna om twee gehele getallen en voer daar de gekozen bewerking op uit. Schrijf het antwoord naar het scherm. Maak deze oefening met switch.
- Ontwerp een consoletoepassing waarmee je de namen van de dagen van de week omzet in cijfers (maandag is 1, dinsdag 2,...). Gebruik een switch. Gebruik de default om foute invoer op te vangen.

3. Ontwerp een consoletoepassing waarmee je de maandnummers omzet in maandnamen (1 is januari, 2 februari,...). Gebruik een switch. Vang uiteraard alle mogelijke foute invoeren op.
4. Ontwerp een consoletoepassing waarmee je op basis van de taalkeuze de tekst “Goeiedag” in de gekozen taal weergeeft. Doe dit voor Nederlands, Engels, Frans en Duits. Als de gebruiker opnieuw iets intypt zegt hij nog vaarwel in dezelfde taal.
5. Ontwerp een console toepassing waarmee je berekent tot welke groep iemand behoort: Op basis van de leeftijd wordt de groep bepaald:
  - Jonger dan 5 jaar: gratis;
  - Tussen de 5 en de 12: halve prijs;
  - Tussen de 13 en de 55: vol tarief;
  - 55+: gratis

Gebruik dan een switch om uit te schrijven hoeveel iemand zijn museumpas moet betalen op basis van de groep waarin hij valt.

6. De solden lopen op hun einde en de uitbater van de schoenwinkel geeft speciale kortingen. Ontwerp een consoletoepassing waarmee de uitbater van een schoenwinkel het totaal te betalen bedrag kan berekenen. Als je één paar schoenen koopt, krijg je 20 % korting, bij de aankoop van 2 paar schoenen krijg je 40% korting en bij aankoop van meer dan twee paar schoenen zelfs 60 % korting. Een paar schoenen kost 125 EUR. Gebruik enkel switch, geen if.

## Oefeningen: lussen (loops)

1. Schrijf een consoletoepassing die de eerste 100 even getallen in beeld brengt.
2. Schrijf een consoletoepassing dat de tafel van vermenigvuldiging afdrukt. Vraag aan de gebruiker van welk getal hij de tafel wil, en druk voor dat getal de tafel af t.e.m. x10. Gebruik gehele getallen.
3. Schrijf een consoletoepassing dat 490 keer (7 maal 70) in beeld brengt “Ik vergeef de ander zijn schuld”. Gebruik een geneste while. Probeer ook eens met een for lus. En daarna met een geneste for lus (later).
4. We zijn het beu dat gebruikers foutieve input geven. Vraag de gebruiker om een getal in te geven. Blijf hem vragen om een getal, zolang hij iets anders ingeeft. Als hij een getal

heeft ingegeven, bedank je hem door alle getallen van 0 tot zijn getal onder elkaar te tonen.

5. Schrijf een consoletoepassing dat deze output in het console venster laat zien:

```
*  
**  
***  
****  
*****  
*****
```

Gebruik een geneste lus, probeer zowel met while als for. Schrijf elk sterretje apart uit.

6. Schrijf een consoletoepassing dat deze output in het console venster laat zien:

```
0****  
01***  
012**  
0123*  
01234
```

Of probeer dit:

```
****1  
***22  
**333  
*4444  
55555
```

Of probeer dit:

```
*  
***  
*****  
*****  
*****
```

Tip: gebruik ook weer een geneste while/for.

7. Schrijf een console applicatie die de gebruiker vraagt om een geheel getal groter dan 2 en de faculteit van dat getal teruggeeft. De faculteit van een getal is het getal vermenigvuldigd met elk kleiner geheel getal t.e.m. 1.  
vb:  $4! = 4 \times 3 \times 2 \times 1$  en  $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$
8. Dit is een uitbreiding op oefening 3.7: Schrijf een programma dat de gebruiker om 2 jaartallen vraagt  $> 0$ . En toon de gebruiker alle schrikkeljaren die tussen die 2 jaartallen inzitten. Zorg ook dat je foutieve invoer opvangt.

9. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Vervolgens krijgt de gebruiker de keuze uit de volgende drie opties: a. voer nog een getal in, b. begin opnieuw (zet de som op 0) of c. beëindig het programma.
10. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Het programma stopt als de gebruiker geen geldig geheel getal invoert. Gebruik hiervoor een `do while`.
11. Schrijf een kassaprogramma. De gebruiker moet eerst kiezen wat hij wil aanrekenen: (1) een maaltijd (€10), (2) een drankje (€1.5), (3) een dessert (€4) of (4) niets meer. Druk de som af van de tot dan gekozen aangerekende items. Blijf vragen naar een nieuw item tot de gebruiker "niets meer" kiest. Wanneer de som groter is dan €30, geef dan een korting van 10% op de totale aangerekende prijs.
12. Schrijf een programma dat berekent of een ingegeven getal een priemgetal is of niet. Een priemgetal is enkel deelbaar door 1 en zichzelf.
13. Schrijf een programma dat de eerste 30 fibonacci-getallen afdrukt. De fibonacci-reeks is een reeks die begint met 2 x een 1, en elk volgend getal is de som van de twee vorige. Dus de reeks gaat als volgt: 1 1 2 3 5 8 13 etc. Zie [https://en.wikipedia.org/wiki/Fibonacci\\_number](https://en.wikipedia.org/wiki/Fibonacci_number) voor meer info.
14. Schrijf een consoletoepassing dat de tafels van vermenigvuldiging afdrukt. Vraag aan de gebruiker tot welk getal hij wil vermenigvuldigen en maak voor elk getal tot en met dat getal de tafel t.e.m. x10.
15. Schrijf een programma dat een string vraagt en elk karakter apart uitschrijft, samen met zijn unicode waarde. Doe met een lus die gebruik maakt van de lengte van een string.
16. Schrijf een applicatie die een string vraagt en elk teken omzet met een roman cypher verschuiving van 5 karakters naar rechts. Doe met een lus die gebruik maakt van de lengte van een string.
17. Herschrijf oefening 5 en 6, maar vraag aan de gebruiker voor hoeveel rijen hij dit wil doen en pas de uitvoer ook aan zodat dit gaat voor alle ingevoerde getallen van 1 t.e.m.

## Extra oefeningen lussen (in English)

1. Write a program in C# Sharp to display the first 10 natural numbers (using a while, then using a for loop).

*Expected Output :*

1 2 3 4 5 6 7 8 9 10

2. Write a C# Sharp program to find the sum of first 10 natural numbers.

*Expected Output :*

The first 10 natural number is :

1 2 3 4 5 6 7 8 9 10

The Sum is : 55

3. Write a program in C# Sharp to display n terms of natural numbers and their sum.

Test Data : 7

*Expected Output :*

The first 7 natural number is :

1 2 3 4 5 6 7

The Sum of Natural Number upto 7 terms : 28

4. Write a program in C# Sharp to read 10 numbers from keyboard and find their sum and average.

Test Data :

Input the 10 numbers :

Number-1 :2

...

Number-10 :2

*Expected Output :*

The sum of 10 no is : 51

The Average is : 5.100000

5. Write a program in C# Sharp to display the cube of the number upto given an integer.

Test Data :



Input number of terms : 5

*Expected Output :*

Number is : 1 and cube of the 1 is :1

Number is : 2 and cube of the 2 is :8

Number is : 3 and cube of the 3 is :27

Number is : 4 and cube of the 4 is :64

Number is : 5 and cube of the 5 is :125

6. Test Data :

Input the number (Table to be calculated) : 15

*Expected Output :*

15 X 1 = 15

...

...

15 X 10 = 150

7. Write a program in C# Sharp to display the multiplication table vertically from n to 1.

Test Data :

Input upto the table number starting from 1 : 8

*Expected Output :*

Multiplication table from 10 to 1 up to the number 8

1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80

1x9 = 9, 2x9 = 18 ..... 8x9 = 72

...

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

.

8. Write a program in C# Sharp to display the n terms of odd natural number and their sum.

Test Data

Input number of terms : 10

*Expected Output :*

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

9. Write a program in C# Sharp to display the n terms of odd natural number and their sum.

Test Data

Input number of terms : 10

*Expected Output :*

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

## Oefeningen: Arrays

1. Schrijf een programma dat de gebruiker tien getallen in laat voeren en dat vervolgens die getallen in omgekeerde volgorde terug schrijft naar het scherm. Maak gebruik van een array om de invoer op te slaan. Pas daarna je programma aan om dit voor 20 getallen te doen. Als je dit goed gedaan hebt, hoeft je maar op 1 plaats iets aan te passen. Het resultaat geef je 2x weer in de console, 1x met een for loop en 1x met een foreach loop.
2. Ontwerp een console applicatie waarmee je aan de gebruiker 5 kommagetallen opvraagt. Bewaar de getallen in een array. Vervolgens vraag je aan de gebruiker het hoeveelste getal hij wil zien. Het resultaat geef je weer in de console.
3. Schrijf een programma dat een array genereert van 20 random getallen van 1 tot en met 5. Vraag aan de gebruiker een getal tussen 1 en 5 (inbegrepen) en tel hoeveel keer dit getal voorkomt in de array. Print het aantal af.
4. Ontwerp een console waarmee je aan de gebruiker 10 getallen opvraagt. Je toont als resultaat in de console enkel de even getallen die de gebruiker heeft ingegeven.
5. Schrijf een programma dat nakijkt of een ingegeven string een palindroom is. Een palindroom is een woord dat hetzelfde blijft als je het omdraait (bv negen of redder). Je kunt een string op dezelfde manier gebruiken als een array om elk individueel karakter uit een string op te vragen (dus je kunt `text[2]` doen om het derde karakter van de string op te vragen).
6. Schrijf een programma dat aan de gebruiker vraagt hoeveel getallen hij wil ingeven. Vraag hem dan om dat aantal getallen en sla deze op in een array. Druk het gemiddelde, het minimum en het maximum van deze getallen af. Doe dit eerst zonder hulp van de Array klasse methodes en daarna ook een keer door gebruik te maken van `Array.Sort`

7. Ontwerp een console waarmee je aan de gebruiker 4 namen van studenten opvraagt. Breng die dan gesorteerd in beeld.
8. Pas oefening 6 aan zodanig dat de **getallen** van klein naar groot worden weergegeven.
9. Pas oefening 6 aan zodanig dat de getallen van groot naar klein worden weergegeven.
10. Je declareert 3 arrays met telkens 5 waarden. De eerst 2 arrays vul je op met random getallen tussen 0 en 10. De derde array vul je achteraf op met de som van de overeenkomstige velden uit array 1 en 2 (dus op index 0 van de derde array staat de som van het eerste element uit de eerste array + het eerste element van de tweede array, enz..). Je toont in de console de 3 arrays in de vorm van een som.
11. Ontwerp een consoletoepassing waarmee je de namen van de leerlingen van je klas opvraagt. Bewaar de namen in een array. Je vraagt de gebruiker eerst hoeveel waarden hij wil ingeven en je voorziet dan voldoende ruimte in de array. Toon de namen in de console.
12. Pas de oplossing van oefening 11 aan zodat de namen alfabetisch weergegeven worden in de console.
13. Maak een programma dat eerst aan de gebruiker vraagt hoeveel getallen hij wenst in te voeren. Na de invoer van deze getallen brengt het programma de grootste ingevoerde waarde (samen met zijn positie in de tabel) op de console. Let op: indien alle ingevoerde getallen negatief zouden zijn, mag niet op het scherm komen dat 0 de grootste ingevoerde waarde is.
14. Maak een programma dat een array van 10 elementen opvult met de waarden 1 tot en met 10 :

index	0	1	2	3	4	5	6	7	8	9
value	1	2	3	4	5	6	7	8	9	10

Breng deze getallen 1 voor 1 op de console.  
VERSCHUIF nu de inhoud van de elementen naar onderstaand schema : ( let op: verschuiven en dus niet opnieuw genereren)

index	0	1	2	3	4	5	6	7	8	9
value	2	3	4	5	6	7	8	9	10	1

Breng nu ter controle ook deze elementen 1 voor 1 op de console.
15. Schrijf een programma dat op een grid van 8x10 op 5 willekeurige plekken “bommen” plaatst. Gebruik hiervoor een 2-dimensionele array van booleans. Vraag vervolgens aan de gebruiker om een locatie op de grid (dus vraag om 2 getallen: x en y) en kijk of er een bom op die locatie ligt. Als er een bom ligt, is het spel afgelopen en druk je het aantal

gokken af die hij heeft moeten doen om een bom te vinden. Zorg dat je applicatie nooit crasht, ook niet bij foute invoer

16. Breid oefening 15 uit zodat de speler een waarschuwing krijgt als hij “vlakbij” een bom zit maar er nog niet op. Kijk hiervoor in alle vakjes rond de gekozen locatie (dus als je 2,2 gekozen hebt, kijk dan of er een bom zit in 1,1 of 1,2 of 1,3 of 2,1 of 2,3 of 3,1 of 3,2 of 3,3). Tip: let erop dat je niet over de grenzen van de array gaat (-1,0 is geen geldige locatie!).

17. Maak een console applicatie die een boek kaarten in een array myCards (type string) heeft. Harten negen kort je af H9 enz. Trek nu een willekeurige kaart uit het boek telkens de speler op enter duwt. Als de speler “fold” ingeeft, dan stop de applicatie.

Breek deze oefening op in deelproblemen en werk elk deelprobleem afzonderlijk af.

1. Maak een array van 52 kaarten
2. Trek willekeurige kaart als je op enter duwt, tot in het oneindige
3. Stop als je “fold” typt
4. Onhoud in tweede array welke kaart al getrokken is (maar doe hier voorlopig niets mee)
5. Controleer als je een kaart trekt of die al getrokken is en dus in de tweede array -steekt. Trek een nieuwe als dit het geval is (tot in het oneindige)
6. Als er 52 kaarten zijn getrokken, begint alles opnieuw.

Tips:

Gebruik duidelijke namen (arrDeckOfCards, arrDeckOfCardsPulled, mustPullAgain, cardAlreadyPulled, ...). IN HET ENGELS!!!!

Gebruik veel Console.WriteLine om duidelijk te maken wat er aan het gebeuren is (de hoeveelste kaart wordt er getrokken bijvoorbeeld)

18. Breid oefening 17 uit zodat je nooit een kaart kan trekken die al getrokken is. Dus na 52 keer een willekeurige kaart trekken, moet alles opnieuw starten. Tip: hou bij in een tweede array of een kaart al getrokken is of niet. Deze oefening kan je op veel verschillende manieren oplossen.

19. Breid oefening 18 uit zodat de boek kaarten geschud wordt. Als de speler “shuffle” ingeeft, dan wordt er geschud. Controleer of je dit goed hebt gedaan door te debuggen en zo de inhoud van de array in beeld te brengen.

20. Schrijf een programma dat een ingegeven lijst van getallen sorteert. Druk de gesorteerde lijst af. Gebruik hiervoor geen bestaande functie (zoals Array.Sort) maar implementeer alles zelf.

21. Ontwerp een console applicatie waarmee je aan de gebruiker 5 strings opvraagt. Bewaar ze in een array. Vervolgens vraag je aan de gebruiker de hoeveelste string hij wil

zien. Het resultaat geef je weer in de console. Zorg dat foutieve invoer of een ongeldige index je applicatie niet doet crashen, maar proper afhandelt met een foutmelding.

22. Ontwerp een console applicatie waarmee je aan de gebruiker 5 strings opvraagt. Bewaar ze in een array. Vervolgens vraag je een string en zoekt in de array op welke plaats die evt voorkomt in je array. Als de string niet voorkomt schrijf je uit "string not found". Het resultaat geef je weer in de console. (Gebruik Array.IndexOf)
23. Schrijf een programma dat een array genereert van 20 random getallen van 1 tot en met 5. Vraag aan de gebruiker een getal tussen 1 en 5 (inbegrepen) en tel hoeveel keer dit getal voorkomt in de array. Print het aantal af.
24. Ontwerp een console waarmee je aan de gebruiker 10 getallen opvraagt. Je toont als resultaat in de console enkel de even getallen die de gebruiker heeft ingegeven.
25. Schrijf een programma dat een spelwereld genereert voor een platformspel en toont aan de speler. Sla de wereld op in een 2D array.
  - a. Vraag eerst aan de gebruiker hoe groot de wereld moet zijn (hoogte en breedte)
  - b. Elk element in de array stelt een vakje voor in de wereld. Dit kan zijn:
    - i. Berg
    - ii. Water
    - iii. Gras
    - iv. Bos
    - v. Gebruik hiervoor een zelfgemaakt enum datatype.
  - c. Elk veld wordt random ingevuld, maar de rand (boven, onder, links en rechts) is altijd berg, zodat de speler niet uit het veld kan lopen.
  - d. De wereld wordt niet met de namen van de velden getoond, maar 1 letter per veldje (berg = M, water = ~, Gras is i, bos is F van Bos)
  - e. Als de wereld getoond is vraag je of de gebruiker tevreden is met de wereld of, of hij een nieuwe wil genereren (de grootte blijft gelijk) en voer je zijn wensen uit.
26. Schrijf een blad-steen-schaar spelletje. Vraag eerst de naam van de speler. De computer vraagt: daarna in een lus 'R'(ock), 'P'(aper), 'S'(cissors), 'A'(play again) of 'E'(xit) aan de gebruiker. Hij genereert een zelf random antwoord (op basis van een getal), en steeds verschillend van het vorige antwoord dat hij gegeven heeft. Je controleert wie er wint en toont de huidige score (in de vorm Computer XX punten, SPELERNAAM XX punten). Bij Play Again zet je de scores terug op 0, bij Exit, schrijf je eerst uit wie de winnaar is.
27. Variatie op oefening 22 uit met een BEST OF 5 concept dus wie het eerste 3x wint is de winnaar. Je speelt dus maximaal 5 ronden. Hou voor elke ronde in een 3D array bij: wat de speler heeft gekozen, wat de computer heeft gekozen en wie de winnaar is ('S' voor speler, 'C' voor computer). Je hoeft niet te vragen of de speler wil stoppen, het spel stopt

automatisch als er 1 winnaar is. Zorg wel dat foutieve invoer wordt opgevangen en de vraag wat de speler kiest wel opnieuw gesteld wordt. Sla de ingevoerde cijfers op in een 2D array (eerste rij zijn de keuzes van speler 1, tweede rij zijn de keuzes van speler 2).

28. Schrijf een programma dat aan de gebruiker vraagt hoeveel rijen en hoeveel kolommen er moeten zijn die een random char bevatten. Vul deze in met random chars (waarden tussen 65 en 91 (A-Z)). Nu ga je de rijen en de kolommen verwisselen, Maak dus een nieuwe 2D array maar met evenveel rijen als de eerste array kolommen heeft en evenveel kolommen als de eerste rijen heeft. Steek nu in de 2e array elk element op de getransponeerde positie (rijnr wordt kolomnr en omgekeerd, zie vb figuur)

A	B	C			A	D	G
D	E	F			B	E	H
G	H	J			C	F	J



29. Microsoft Mijnenveger <https://youtu.be/Z0EAysRluJk>

30. Game of life <https://youtu.be/ouipbDkwHWA>

## Oefeningen: Methodes

1. Schrijf een methode die een gebruiker tekst vraagt en de ingevoerde tekst terug uitschrijft
2. Schrijf een methode die een double getal vraagt, afrond en omzet naar een geheel getal en terug uitschrijft
3.
  - a. Schrijf een methode die tekst vraagt en iedere karakter 3 plaatsen naar links in het alfabet verschuift (dus d wordt a,...) let weer op met de eerste chars. De methode schrijft het resultaat uit
  - b. Schrijf een methode die tekst vraagt en iedere karakter 3 plaatsen naar rechts in het alfabet verschuift (dus a wordt d,...) let weer op met de eerste chars. De methode schrijft het resultaat uit
  - c. Pas beide methodes aan zodat ze het resultaat in plaats van uitschrijven teruggeven (en schrijf het resultaat uit in main)

- d. Pas de methodes aan zodat ze tekst als parameter meekrijgen (die je dan vraagt aan de gebruiker vanuit main.
  - e. Laat zien dat je een vertaling kan maken van ingevoerde tekst en het resultaat terug kan vertalen naar het origineel door beide na elkaar aan te roepen (tweede methode krijgt het resultaat van de eerste als waarde voor de parameter.
  - f. Doe nu hetzelfde, maar geef als bijkomende parameter een getal, dat aangeeft hoeveel letters je in het alfabet moet verschuiven en pas je verschuif code aan.
4. Schrijf een methode AskNumber, die een geheel getal aan de gebruiker vraagt en true of false teruggeeft als dit geslaagd is of niet. Gebruik een out-parameter om het getal terug te geven. Het resulterende getal is 0 als de gebruiker geen geldige invoer heeft ingegeven. Deze methode kan je ook in verdere oefeningen hergebruiken. Test door vanuit je main deze methode op te roepen en het resulterende getal te tonen aan de gebruiker.
5. Schrijf een methode KeepAskingNumber zodat je de vraag om een getal blijft herhalen totdat er een geldig getal wordt ingegeven. Nu geef je de waarde terug als resultaat en dus geen bool meer (want de methode stopt pas als er een geldig getal werd ingegeven). Deze methode heeft geen parameters. Test vanuit je main. Gebruik hiervoor een loop.
6. Herschrijf oefening 4 en 5 zodanig dat dit werkt voor floating point getallen.
7. Maak een nieuw project aan waarin je je code voor KeepAskingNumber uit oefening 6 kopieert. Maak een array aan van 10 integers, loop nu met een lus door je array en vul elk element in m.b.v. je KeepAskingNumber methode. Schrijf daarna met een foreach lus al je elementen van de array uit in je console.
8. Maak een programma dat 5 getallen vraagt aan de gebruiker, en deze optelt en op het einde de som afdruckt. Gebruik KeepAskingNumber zodat je foutief ingegeven getallen opvangt zonder dat het programma crasht.
9. Breid oefening 8 uit zodat je eerst aan de gebruiker vraagt hoeveel getallen hij wil ingeven (dus niet altijd 5).
10. Schrijf een functie IsEven die als parameter, een geheel getal heeft en als resultaat of dit getal even is (of anders gezegd: deelbaar is door 2) of niet. (Boolean). Vraag een getal aan de gebruiker en controleer of dit even is en schrijf dit ook uit vanuit je main methode.
11. Schrijf een methode, AddOne die 1 optelt bij zijn integer parameter en het resultaat teruggeeft. Roep de methode 4x op vanuit je main en toon het resultaat. Van de oproepen, zijn er 2 die rechtstreeks een waarde invoeren als parameter en 2 die een waarde by value doorgeeft vanuit een eerder aangemaakte variabele uit je main.

12. Schrijf een methode `CmToInch`, die bij een gegeven aantal centimeters het aantal inches berekent en retourneert. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Test met meerdere waarden vanuit `Main`.
13. Schrijf een methode `CalculateCubeVolume` die bij gegeven lengte van een ribbe (=zijde van een kubus) het volume van de kubus retourneert.
14. Schrijf een methode `Smallest` die als parameters twee gehele getallen heeft en die als resultaat het kleinste van de twee getallen geeft.
15. Schrijf een methode `TimeDiffInSec` met zes parameters en een integer als resultaat. De invoer bestaat uit twee tijdstippen in uren, minuten en seconden. De methode moet het verschil tussen deze tijdstippen in seconden retourneren. Test met meerdere waarden vanuit `Main`.
16. Schrijf een methode `AppendStrings`, die 3 string parameters meekrijgt en deze achter elkaar plakt. Het resultaat is 1 string die wordt geretourneerd. Test vanuit je `main` door het resultaat van de aanroep rechtstreeks vanuit een `Console.WriteLine` uit te schrijven.
17. Schrijf een methode `SumArray` die een array van gehele getallen als parameter mee krijgt. Zorg ervoor dat het return-resultaat de som is van alle getallen in de array.
18. Maak een array aan en steek er 10 gehele getallen in. Gebruik de `AddOne` methode uit opgave 8 om elk getal uit de array met 1 te verhogen. Doe dit in een `for` lus. Schrijf de array daarna uit met behulp van een `foreach` lus.  
UITBREIDING: return de resultaat array en doe het uitschrijven in de `main`
19. Schrijf een methode `CmArrayToInchArray`, die bij een gegeven array van cms het aantal inches berekent, in een nieuwe array plaatst en die array retourneert. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Of gebruik de methode `CmToInch` uit opgave 9. Test met een array die je maakt in je `Main`.
20. Schrijf een methode `StringToArray` die een string als parameter krijgt, en een array van strings teruggeeft. Elke string in deze array bevat exact 1 karakter uit de input-string. Gebruik hiervoor `Substring`.
21. Schrijf een functie `HowManyAreEven` die als parameter, een array van gehele getallen meekrijgt en als resultaat teruggeeft hoeveel van de getallen die erin zitten (mag random zijn) even zijn(of anders gezegd: deelbaar is door 2) of niet. Maak in je `main` een array met random 10 getallen en test je methode.
22. Schrijf een methode, `AddOneToArray` die 1 optelt bij alle elementen van een integer array parameter. Als resultaat return je ook de som van alle nieuwe getallen in de array. Roep de methode 4x op vanuit je `main` en toon het resultaat. Roep dit op vanuit je `main` met een array van 5 random getallen.



23. Schrijf een methode `CmToInch`, die bij een gegeven aantal centimeters het aantal inches berekent en in een tweede out parameter opslaat. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Test met meerdere waarden vanuit `Main`.
24. Maak een array aan en steek er 20 random kommagetallen in. Geef de array mee als ref aan een `ArrayTimes2` methode en verdubbel hierin elk getal. Schrijf de array voor en na het uitvoeren van de methode in de main uit m.b.v. een `foreach` lus. Random kommagetallen genereer je m.b.v `Random.NextDouble()`
25. Schrijf een methode `InchToCm`, die een ref parameter heeft met het gegeven aantal inches. Gebruik diezelfde ref parameter om het resultaat van de omrekening naar centimeters op te slaan. Het aantal inches berekent en in een tweede out parameter opslaat. Deel het aantal inches door 0.394 om het aantal cm's te verkrijgen. Test met meerdere verschillende waarden vanuit `Main`.
26. Schrijf een functie `AddOneAndCountEven` die als parameter, een array van gehele getallen meekrijgt en in 2 out parameters opslaat:
- a. Een array met alle elementen van de array -1.
  - b. hoeveel van de originele getallen die erin zitten even zijn
- Maak in je main een array met random 5 getallen en test je methode.
27. Schrijf een methode, `AddOneToArray` die 1 optelt bij alle elementen van een integer array parameter. Als resultaat return je ook de som van alle nieuwe getallen in de array. Roep de methode 4x op vanuit je main en toon het resultaat. Roep dit op vanuit je main met een array van 5 random getallen.
28. Vraag aan de gebruiker hoeveel getallen hij in een array wil steken. Steek in een array er evenveel random kommagetallen in. Geef de array mee als ref aan een `ArrayTimes2` methode en verdubbel hierin elk getal. Schrijf de array voor en na het uitvoeren van de methode in de main uit m.b.v. een `foreach` lus. Random kommagetallen genereer je m.b.v `Random.NextDouble()`.
29. Maak een console applicatie die een boek kaarten in een array `myCards` (type string) genereert heeft. Maak gebruik van een methode `GenerateDeck`, die een volledig kaartspel (52 kaarten genereert). Deze gebruikt een tweede methode `GenerateCardString` die 2 elementen omzet in de naam van de kaart (eerste parameter is een integer die de soort representeert, tweede parameter representeert de waarde 1 - koning = 13). Schrijf na de methode `GenerateDeck` het kaartspel uit.

# Oefeningen: Enums, Lists en andere datatypen

1. Maak een console applicatie die een boek kaarten in een list myCards (type string) heeft. Harten negen kort je af H9, ruiten boer met R11, schoppen aas het S1, klaveren koning met K13 enz. Genereer deze lijst met for-loops. Trek nu een willekeurige kaart uit het boek telken de speler op enter duwt. Als de speler "fold" ingeeft, dan stop de applicatie.
2. Breid oefening 1 uit zodat je nooit een kaart kan trekken die al getrokken is. Dus na 52 keer een willekeurige kaart trekken, moet alles opnieuw starten. Deze oefening kan je op veel verschillende manieren oplossen.
3. Pas vorige oefening aan, gebruik een Enum Suit om je type kaart te specificeren.
4. Breid oefening 2 uit zodat de boek kaarten geschud wordt in een methode. Controleer of je dit goed hebt gedaan door te debuggen en zo de inhoud van de list in beeld te brengen. Maak hiervoor een methode die een boek kaarten krijgt en waar een andere boek kaarten uit komt die geschud is.
5. Programmeer op basis van vorige oefeningen een vereenvoudigd blackjack spel. Elke kaart krijgt gewoon zijn waarde, een prentje = 10 en een aas is altijd 1. Je wint het spel als je exact 21 kan halen. Je vraagt de speler telkens of hij nog een kaart wil. Na elk spel toon je of de speler gewonnen heeft of verloren en vraag je of hij nog een keer wil spelen.
6. Schrijf een programma dat de gebruiker tien getallen in laat voeren en dat vervolgens die getallen in omgekeerde volgorde terug schrijft naar het scherm. Maak gebruik van een list om de invoer op te slaan. Pas daarna je programma aan om dit voor 20 getallen te doen. Als je dit goed gedaan hebt, hoef je maar op 1 plaats iets aan te passen.
7. Ontwerp een console applicatie waarmee je aan de gebruiker 5 getallen opvraagt. Bewaar de getallen in een list. Vervolgens vraag je aan de gebruiker het hoeveelste getal hij wil zien. Het resultaat geef je weer in de console en verwijder je uit de lijst.
8. Ontwerp een console waarmee je aan de gebruiker getallen laat opvragen tot hij stop intypt. Je slaat de getallen op in een lijst. Je toont als resultaat in de console enkel de even getallen, in gesorteerde volgorde van groot naar klein.

9. Ontwerp een console waarmee je aan de gebruiker 4 namen van studenten opvraagt en opslaat in een lijst. Breng die dan gesorteerd in beeld.
10. Pas oefening 7 aan zodanig dat de **getallen** van klein naar groot worden weergegeven.
11. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Vervolgens krijgt de gebruiker de keuze uit de volgende drie opties: a. voer nog een getal in, b. begin opnieuw (zet de som op 0) of c. beëindig het programma. Als het programma beëindigd wordt schrijf je nog 1x de hele lijst van getallen uit als 1 grote som en ook het resultaat van die som.
12. Maak een programma dat een list opvult met de waarden 1 tot en met 10 :
- |       |   |   |   |   |   |   |   |   |   |    |
|-------|---|---|---|---|---|---|---|---|---|----|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  |
| value | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
- Breng deze getallen 1 voor 1 op de console.  
VERSCHUIF nu de inhoud van de elementen naar onderstaand schema : ( let op: verschuiven en dus niet opnieuw genereren)
- |       |   |   |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| value | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
- Breng nu ter controle ook deze elementen 1 voor 1 op de console.
13. Schrijf een programma dat een lijst maakt van 10 willekeurige Shapes. Een Shape is een Circle, een Triangle, een Rectangle of een Square. Daarna schrijf je met een lus de types uit de lijst uit met daarachter de formule om hun oppervlakte te berekenen.
14. Schrijf een programma dat een enum aanmaakt voor de vier windrichtingen. Bereken een random getal die de windrichting bepaald. De hoek het dichtste bij 1 van de hoofdrichtingen (noord = 0° of 360°, oost = 90°, zuid = 180°, west = 270°) wordt dan getoond. Herhaal tot de gebruiker stop zegt.
15. Schrijf een programma dat de gebruiker om 5 namen vraagt. Voeg deze toe aan een lijst van strings. Druk de hele lijst af met een loop als de 5 namen ingegeven zijn.
16. Schrijf een programma dat de gebruiker om een naam vraagt. Voeg deze naam toe aan de lijst. Blijf om nieuwe namen vragen tot de gebruiker STOP typt. Druk dan de volledige lijst af.
17. Breid het programma uit oefening 16 uit. Nadat de gebruiker STOP heeft gedrukt, vraag je opnieuw naar een naam. Maar deze keer VERWIJDER je de naam uit de lijst. Druk daarna de lijst met de overgebleven namen af.

18. Maak een Enum Language aan die drie elementen bevat: Dutch, French en English. Gebruik een random getal om één van de drie Enum-opties te kiezen, en groet de gebruiker in deze taal.
19. Maak een applicatie die een enum RainbowColors definieert. De kleuren zijn red, orange, yellow, green, blue, indigo, violet. Geef elke kleur ook het nummer dat overeenkomt met zijn positie in de regenboog (dus red = 1,...). Vraag nu aan de gebruiker een kleur als naam, komt de kleur niet voor in de regenboog, dan schrijf je uit, This color is not a part of the rainbow. Anders Schrijf je uit, This color is de Xth color in the rainbow (vervang X door de waarde van de kleur). (hint: gebruik Enum.TryParse)
20. Maak een applicatie die een enum TemperatureFeeling heeft voor hot, warm en cold. Het is cold als de temperatuur kleiner is dan 15 graden, hot als het warmer is dan 30 graden en anders warm. Gebruik de waarden die in de enum overeenkomen met de namen als grenzen voor je if selecties (zodat we makkelijk ook hot kunnen definiëren als 35 graden en er niets anders aan de applicatie verandert). Geef voor 10 random getallen aan of ze cold, warm of hot zijn. Maak een methode die op basis van dat getal een TemperatureFeeling teruggeeft

## Oefeningen: Klassen

1. Ga naar de online Tower Defense game Bloons Tower Defense 5 (<https://ninjakiwi.com/Nederlands/Games/Tower-Defense/Play/Bloons-Tower-Defense-5.html#.XFF97IxKhPY>). Ga na wat de mogelijke klassen zijn en wat de objecten in onderstaande screenshot. Welke eigenschappen en methoden zouden de klassen kunnen hebben. Hebben alle objecten van een klasse dezelfde eigenschappen?



- 2.
3. Maak een klasse Mission aan, die de missie van een spel zal bijhouden. Deze Missie klasse houdt volgende gegevens bij:
  - a. Een missienaam
  - b. introtext
  - c. een status (ToDo, InProgress, Failed, Completed) => gebruik enum
  - d. pointsWhenCompleted
  - e. outro text

Daarnaast heeft de klasse

- een methode Complete, die geeft een string terug bestaande uit de outro text en heeft een out parameter, waarin wordt ingevuld hoeveel aan de score zal worden toegevoegd. Verander de status dan ook naar Completed.
- voor elk gegeven properties.
- een methode StartMission, die de status verandert naar InProgress, deze retournt de introtext
- een methode Fail, die de status verandert naar Failed

Maak in je main methode twee missies aan, vul alle gegevens in. en start de missie, complete 1 missie en fail 1 missie. Toon alles wat er gebeurt proper in de console en check met debuggen/steppen door je code of alles lukt.

4. Ontwerp een klasse voor de opslag van de gegevens van leerlingen. Sla de voornaam, de naam en het geboortjaar elk op in een objectvariabele. Voeg voor elk gegeven

properties toe en maak een methode die alle gegevens als text teruggeeft. Maak in je hoofdprogramma 5 leerling objecten aan (met verschillende gegevens) en schrijf hun data uit. Maak ook een methode GetFullName die de volledige naam teruggeeft als hulpmethode (intern in de klasse) en schrijf deze uit voor elk van de klasse-objecten.

5. Ontwerp een klasse voor de opslag van de gegevens van een rechthoek. Sla de breedte en de hoogte elk op in een objectvariabele. Maak een methode die de oppervlakte van de rechthoek als waarde teruggeeft. Gebruik deze in de main om het resultaat weer te geven. Breid de oefening uit om 1000 rechthoeken met willekeurige breedte en hoogte aan te maken en geef de gemiddelde oppervlakte weer in het console venster. Verwijder daarna alle rechthoeken uit de lijst met een oppervlakte kleiner dan 10.
6. Ontwerp een klasse CD voor de opslag van de titel en de uitvoerder. De constructor (als je dat al kan) maakt een nieuwe CD aan met de ingegeven titel en uitvoerder. Voeg twee methoden toe om de inhoud van de velden titel en uitvoerder op te vragen. Voeg nog een veld toe waarin een lijst van liedjes staat. Zorg er ook voor dat je een methode hebt waarmee je een liedje kan toevoegen. Voeg een methode toe om alle informatie over de titel, uitvoerder en de lijst van liedjes omvormt naar tekst. Maak in je Main een lijst van CD's. Maak een Shuffle methode die alle nummers van alle CD's oplijst in willekeurige volgorde. Voorzie een duidelijke weergave van de informatie in je console.
7. Ter promotie van Belgische appels en peren wordt een nieuw type fruitautomaat gelanceerd. Maak een nieuwe klasse Fruitautomaat aan. Een fruitautomaat bevat appels en peren. Het maximum aantal van elk is gelimiteerd op 50 stuks.  
Voeg vervolgens een aantal methoden toe om:
  - appels aan te vullen;
  - peren aan te vullen;
  - appels te verkopen;
  - peren te verkopen;
  - te kijken hoeveel stukken fruit er nog in de automaat zitten;
  - te controleren of een bepaalde fruitsoort uitverkocht is.Houd er rekening mee dat je bijvoorbeeld geen appel kan verkopen als er geen appels meer in de automaat zitten.
8. Ontwerp een klasse StudentData voor de opslag van de naam en punten (in %) van een student. Maak properties aan om zowel de naam als de punten in te geven. Voeg een ToString-methode toe die de naam en punten van de student afdruckt. Maak 3 verschillende variabelen aan van type StudentData, vul ze op met verschillende gegevens, en gebruik de ToString-methode om de info af te printen.
9. Voeg een methode GetGrade toe die de graad teruggeeft van een student. Dit zijn de mogelijkheden:

- < 50: niet geslaagd;
- tussen 50 en 68: voldoende;
- tussen 68 en 75: onderscheiding;
- tussen 75 en 85: grote onderscheiding;
- > 85: grootste onderscheiding.

Druk naast de naam en de punten ook de graad af van elke student. In het Engels:

failed,  
sufficient,  
distinction,  
great\_distinction,  
greatest\_distinction

10. Maak een extra methode `IsFailed` aan die `true` teruggeeft als de student < 50% heeft en `false` als de student wel geslaagd is. Druk dit ook af.
11. Gebruik nu een array in plaats van losse variabelen om de `StudentData` op te slaan. Vraag eerst aan de gebruiker hoeveel studenten hij wil ingeven. Vraag vervolgens de naam en de punten van elke student in een loop, en vul zo de array op. Druk uiteindelijk dezelfde informatie af van elke student als in oefening 7.
12. Doe hetzelfde als in oefening 9, maar nu druk je de geslaagde en gefaalde studenten apart af. Dus je drukt eerst af "Deze studenten zijn geslaagd:", gevolgd door alle geslaagde studenten in de array. Daarna druk je af "Deze studenten zijn gefaald:", gevolgd door alle gefaalde studenten. Gebruik hiervoor 2 aparte loops.

## Oefeningen: Overloading & Constructors

1. Maak 3 methodes voor het uitprinten van data met de naam `PrintData`. Eenmaal voor tekst, eenmaal voor gehele getallen en eenmaal voor kommagetallen. Deze printen respectievelijk `TEXT : + parameter`, `INT + parameter`, `FLOAT + parameter` uit in de console. Roep deze op vanuit je `main` met verschillende parameters, check m.b.v. debugging en *step into* dat de juiste methode wordt opgeroepen.

2. Maak methodes voor alle combinaties van het vermenigvuldigen van double en int. Dus een `int * int`, `double * int`, `int * double`, `double * double`. Het resultaat is altijd een double. Kan je alle methodes aanroepen? Probeer dit vanuit je main.
3. Ontwerp een klasse Fraction voor de opslag van de gegevens van een breuk, namelijk teller (enumerator) en noemer (denominator) (beide ints). Voorzie 2 constructors. De eerste constructor maakt een nieuwe breuk aan met standaardwaarden (bv 1/1). De tweede constructor voorziet parameters voor het ingeven van teller en noemer. Voorzie een ToString override die de breuk afdruckt. Voorzie een methode om het quotiënt uit te rekenen. Test voor verschillende breuken. Voorzie alle klassevariabelen van een property (en zorg ervoor dat niemand het cijfer 0 in de noemer kan invullen).
4. Voeg een methode Add toe aan de Fraction-klasse, die een teller meekrijgt, en deze teller optelt bij de teller in je Fraction-object. Overload deze methode zodat je ook een teller en een noemer kunt meegeven. Je moet dan eerst de noemer van beide breuken gelijkstellen om de tellers te kunnen optellen (zie wiskunde lagere school <https://www.youtube.com/watch?v=zMJ3zWPffG8>). Voeg tenslotte een derde overloaded methode Add toe die een ander Fraction-object krijgt, en tel deze bij elkaar op op dezelfde manier als bij de vorige methode. Test met verschillende waarden en overloaded methodes, en print telkens het resultaat af.
5. Maak een klasse Color, die intern 3 integers bijhoudt, waarden tussen 0 en 255, om een kleur op te slaan voor rood/groen/blauw. Zorg dat deze 2 constructors heeft waarmee de waarden kunnen ingevuld worden. De eerste neemt 3 integer variabelen aan als parameter, die gebruikt worden om de klassevariabelen in te vullen. De 2e neemt float variabelen aan (die van waarden tussen 0 en 1 worden omgevormd naar integers tussen 0 en 255 <https://www.youtube.com/watch?v=W5g-TiE6aZk>). Daarnaast zijn er ook 2 methoden ChangeColor, ook met een int en een float versie. Maak een methode ToString die de waarden als string teruggeeft. Test vanuit je main meerdere MyColor objecten.  
Extra: Maak een enum PredefinedColors met daarin de kleuren rood (255,0,0), Green (0,255,0) en Blue (0,0,255). Zorg dat je aan je constructor en je changecolor ook een PredefinedColor variabele kan meegeven.
6. Maak voor elke klasse-oefening in deel 8 een constructor die op basis van parameters elke klassevariabele initialiseert. Wat gebeurt er bij het aanmaken van je objecten van die klasse? Los dit op door:
  - a. Eerst: een default constructor (zonder parameters) zelf te overloaden die alle klassevariabelen op default waarden initialiseert.
  - b. Daarna: door het aanmaken van de objecten te vervangen door oproepen van deze nieuwe constructor (zet de default constructor even in commentaar)



7. Maak een klasse `ColorConsole` die als constructor 2 parameters aanneemt: de achtergrondkleur en de voorgrondkleur (de kleur van de tekst). Beide parameters zijn van type **`ConsoleColor`**. Dit is een enum die je reeds gebruikt hebt in de eerste oefeningenreeks, maar toen wist je nog niet dat dit een enum was. Sla beide parameters op in een membervariabele. Voeg een methode `PrintColors` toe die afdruckt wat de opgeslagen kleuren zijn. Een variabele van type **`ConsoleColor`** kun je converteren naar een string door er de **`ToString()`**-methode op op te roepen. Dit kan bij elke enum! Test dit door een `ColorConsole`-object aan te maken en de `PrintColors`-methode op te roepen. Onthoud: je kunt een enum meegeven als parameter door bv **`ConsoleColor.Red`** te typen.
8. Voeg een methode `WriteLine` toe die een string als parameter neemt, en de string afdruckt met als voorgrond- en achtergrondkleur de kleur uit de membervariabelen. Dit kun je doen door **`Console.ForegroundColor = color`**, **`Console.BackgroundColor = color`**, en **`Console.WriteLine()`** te gebruiken.
9. Voeg een methode `ChangeColors` toe die, net zoals de constructor, de achtergrond- en voorgrondkleur meekrijgt als parameters, en deze opslaat in de membervariabelen van de klasse. Gebruik deze methode om verschillende keren `WriteLine` op te roepen met verschillende achtergrond- en voorgrondkleurencombinaties.
10. Tenslotte, voeg een overloaded versie van `WriteLine` toe aan `ColorConsole` die als parameters een string en twee `ConsoleColor`-parameters binnenkrijgt. Druk deze string af met de achtergrond- en voorgrondkleur die je als parameter hebt meegegeven.
11. Pas beide overloaded `WriteLine`-methodes aan zodat je onthoudt wat de originele kleuren waren van de console (dus de originele waarden van **`Console.ForegroundColor`** en **`Console.BackgroundColor`**). Deze waarden zet je na het oproepen van **`Console.WriteLine()`** terug. Test dit door een aantal keer `WriteLine` op te roepen op een `ColorConsole`-object en dan een gewone klassieke `Console.WriteLine`. Deze zou afgedrukt moeten worden in de oorspronkelijke klassieke kleuren van de console.

## Oefeningen: Text File I/O

1.
  - a. Maak een klasse `ConsoleSettings` hierin maak je variabelen voor
    - i. Voorgrondkleur ( als `ConsoleColor`)
    - ii. Achtergrondkleur ( als `ConsoleColor`)

- iii. Aantal opstartbeeps
- b. In je hoofdprogramma
  - i. Zet een ConsoleSettings instantie om naar een json string
  - ii. Schrijf die json string weg naar een tekstbestand
- c. In je hoofdprogramma
  - i. Check of je de file consolesettings.json kan lezen
  - ii. Indien ja
    - 1. Lees de info uit file en maak een consolesetting variabele op basis van de ingelezen data
    - 2. Stel de consolesettings in, speel de beeps en toon "Hello Word" in de opgeslagen kleuren aan de gebruiker
  - iii. Vraag aan de gebruiker een voorgrond en achtergrondkleur en aantal beeps en sla ze op in een "consolesettingsFromUser.json" bestand en toon het resultaat zoals in ii "Indien ja".
- d. Variatie B:
  - i. Maak een lijst van verschillende ConsoleSettings. Sla ze op in een file, lees ze allemaal terug in en toon ze via een ToString aan de gebruiker
  - ii. Laat de gebruiker 1 van de opties kiezen op basis van een nummer (1 is de eerste...)
  - iii. Toon "Hello World" met de gekozen setting
- e. Variatie C:
  - i. Maak een bijkomende klasse MultipleSettings
  - ii. Hierin hou je een List bij van ConsoleSettings (en een property die er toegang toe geeft)
  - iii. Hierin heb je een activeSetting (dit is een index die verwijst naar de hoeveelste setting actief is)
  - iv. Voeg een AddSetting toe die een nieuwe Setting toevoegt
  - v. Zorg dat je MultipleSettings kan wegschrijven en inlezen
  - vi. Maak een menu waarin je kan kiezen
    - 1. Show active settings
    - 2. Set active setting
    - 3. Add new setting
    - 4. Save MultipleSettings object
    - 5. Stop applicatie

En werk alles proper uit

2.

- a. Maak een klasse Animal met hierin
  - i. Soort
  - ii. Aantalpoten
  - iii. Een constructor
  - iv. over schrijf de methode ToString, die de animal data in tekstvorm teruggeeft

- v. Maak in je Program.cs een methode WriteAnimalsToFile die een animal array meekrijgt als parameter en deze naar je desktop folder animal.txt schrijft.
  - b. Maak in je main een array van 5 Animal objecten en schrijf deze naar file. Check of de file ok is
  - c. Maak een methode readAnimals in je Program.cs die een filenaam meekrijgt. Deze methode leest alle animal data terug in en retournt een array van Animals met de waarden die overeenkomen met de uit tekst gehaalde animal objecten. Schrijf deze array terug uit in je main
- 3.
- a. :Maak een klasse Order met hierin
    - i. Tafelnummer
    - ii. Lijst met gerechten
    - iii. Lijst met dranken
    - iv. over schrijf de methode ToString, die de order data in tekstvorm teruggeeft.
  - b. Vraag in je main of de gebruiker naar een file wil schrijven of uit een file wil lezen. Als de gebruiker kiest voor schrijven maak je een order en schrijf je deze weg naar een file. Als de gebruiker kiest voor lezen lees je de file uit en schrijf je de order uit in de console met behulp van de toString methode. Wanneer er geen file is of niets in de file staat schrijf je "No orders"
  - c. Als stap a en b lukken kan je deze oefening uitbreiden door enums te gebruiken voor eten (Spaghetti, Vol au vent, Hamburger en Meatballs) en drinken (Water, Cola, Ice tea, White wine, Red wine en beer).

## Oefeningen: Collections

- 1.
- a. Vraag aan de gebruiker een string text. Voer van de ingelezen tekst elk karakter individueel toe aan een Queue. Schrijf daarna elk element uit de queue terug uit.
  - b. Vraag aan de gebruiker een string text. Voer van de ingelezen tekst elk karakter individueel toe aan een Stack. Schrijf daarna elk element uit de stack terug uit.
  - c. Maak een methode CheckPalindrome, die test of een ingevoerde string een palindroom is (woord dat van achter naar voor hetzelfde is). Doe dit door elk karakter in een stack en in een queue te steken, daarna check je of de

elementen die je eruit haalt hetzelfde zijn. Zijn ze allemaal gelijk (en evenveel), dan heb je een palindrome (want een stack is LIFO, queue is FIFO)

2. Maak een programma dat checkt of een reeks haakjes uitgebalanceerd is (dus haakjes mooi in volgorde en open en gesloten. Je hebt drie soorten haakjes: (), [], {} die in elkaar geplaatst mogen worden. Je vraagt aan de gebruiker een reeks haakjes. Maak een methode `IsBalanced` die teruggeeft of ze in balans zijn of niet, Gebruik een Stack.  
\*DIT IS EEN STEVIGE DOORDENKER\*
3. Maak een applicatie waar je strings mee inleest (zogezegd url's) en die uitprint als `SHOWING: url`. Je voegt ze ook toe aan een stack. Als de string een gelijk is aan `backspace`, betekent dit dat je terug wil naar de vorige url, dan toon je de vorige url (bovenste element van de stack). Als je `backspace` drukt en de stack is leeg, dan schrijf je uit `SHOWING: HOME`.
4. `HashSet` oefening. Vraag aan de gebruiker een woord, sla alle karakters uit dit woord in een `HashSet` op. Toon ofdat een letter al in de hashset zat als je het toevoegt of niet. Loop door alle chars van a-z en schrijf uit of dit karakter in het ingevoerde woord zat (als het in de hashSet zit).
5.
  - a. Maak een klasse `Bullet`, waarin een integer `damage` zit, voeg een constructor toe die je een `damage` kan meegeven.
  - b. Maak een klasse `Gun`, die een stack `magazine` heeft van `Bullets`, die is oorspronkelijk leeg. Maak een methode `AddBullet`, die een `Bullet` toevoegt aan de `magazine` variabele.
  - c. Maak een methode `Fire`, die je de bovenste bullet van `magazine` teruggeeft en verwijdert.
  - d. Maak een methode `CheckNextBullet`, die returnt de bovenste bullet van `magazine` zonder die te verwijderen.
  - e. In je `main`, maak je een `Gun` aan.
  - f. Als je op `L(oad)` drukt voeg je 5 kogels toe met random `damage` tussen 10 en 100,
  - g. druk je op `F(ire)`, dan schiet je 1 kogel af. Print af, "Shot bullet with damage X" waarin X de `damage` van die kogel is
  - h. Druk je op `C(heckMagazine)`, dan schrijf je uit wat de volgende kogel zou zijn die je afschiet.
  - i. Als extra kan je dit allemaal in 2 lists steken en deze lists van bullets en guns naar file schrijven en terug inlezen.
6.
  - a. Maak een methode die een string meekrijgt en een `HashSet` returnt met daarin alle karakters die voorkomen in de tekst

- b. Vraag aan de gebruiker een string, en voer de methode uit. Check of elk letter in het alfabet voorkomt in de tekst of niet en schrijf dit uit
- c. Vraag aan de gebruiker een filenaam, lees nu heel de tekst uit de file in en doe hetzelfde voor die.
- d. Maak een nieuwe methode, die hetzelfde doet, maar in plaats van individuele karakters slaat hij alle woorden op in de hashset. Splitten van heel je tekst kan bvb zo (enkele alternatieven in comment):

```
//string[] words = text.Split(); //splits on all whitespaces
char[] separators = new char[] { ',', '.', '!', '?', '\\', '\\', ' ', '\\n', '\\t' };
string[] words = text.Split(separators);
//string[] words = Regex.Split(text, @"\s+");
```

- e. Toon alle woorden van de file in de console.
7. Maak een Dictionary aan waarbij een char een key is en een getal als value
    - a. Hermaak nu vorige oefening waarbij je in de dictionary bijhoudt hoe vaak elk karakter voorkomt in een ingevoerde string. Toon in de console
    - b. Hermaak nu vorige oefening waarbij je in de dictionary bijhoudt hoe vaak elk woord voorkomt in een file.
  8. Maak een Dictionary aan waarbij een string een key is en een getal als value
    - a. Lees een tekstfile in via File.ReadAllText
    - b. Split de tekst op basis van spaties
    - c. Hermaak nu vorige oefening waarbij je in de dictionary bijhoudt hoe vaak elk woord voorkomt in een file.

# Geavanceerdere oefeningen

Deze oefeningen lijken op wat we op het examen zullen vragen. Het gaat hem over complexere problemen die wat meer logisch denken vragen. Deel elke opgave op in kleine stukjes en tackle dan stukje per stukje. Dat lukt meestal niet in een keer. Dus ga stap per stap te werk.

## 1 Twitter sentiment analyse

Tweets kan je downloaden naar een csv-file: <https://twitter-sentiment-csv.herokuapp.com/>. Kies zelf een onderwerp, bijvoorbeeld "Trump". We kunnen deze file inlezen en tweet per tweet nagaan of de tekst positief is of negatief (of neutraal). Dat staat op zich ook in de file zelf, maar we willen dit zelf kunnen. We zullen alle tweets analyseren met woorden uit een woordenboek die voorzien zijn van een sentimentscore:

[https://github.com/cjhutto/vaderSentiment/blob/master/vaderSentiment/vader\\_lexicon.txt](https://github.com/cjhutto/vaderSentiment/blob/master/vaderSentiment/vader_lexicon.txt)

Gebruik deze woorden om voor elke tweet een score te berekenen (gebruik de woorden uit de eerste kolom en de score uit de tweede kolom).

Dit kan er dan zo uit zien:



```
D:\Data\02 KdG\04 Programmeren\2018-2019\Programming1\Les\bin\Debug\Les.exe

"RT @EdKrasen: INSTEAD OF GIVING $5 BILLION FOR TRUMP'S WALL\n\n- Fix Flint's water crisis ~ $55 mil\n- Feed all homeless vets 3 meals/day for."
Constains crisis so adds -3,1 to score
Constains giving so adds 1,4 to score
Total score for tweet: -1,7

"RT @EdKrasen: Stop saying 'Donald Trump is not racist.'\n\n'Not racist' people don't joke about mass genocide of another race of people."
Constains joke so adds 1,2 to score
Constains stop so adds -1,2 to score
Total score for tweet: 0

"RT @ezralevin: In 2000
Total score for tweet: 0

"RT @TheUSASingers: I have never been more embarrassed and ashamed of anyone or anything than I am to have Donald Trump be the President of."
Constains ashamed so adds -2,1 to score
Constains embarrassed so adds -1,5 to score
Total score for tweet: -3,6

"RT @tedlieu: Dear @SecPompeo: I look forward to you testifying before the Foreign Affairs Committee. Some questions:\n\n-Why are you ignoring."
Constains dear so adds 1,6 to score
Total score for tweet: 1,6

"ok
Total score for tweet: 0
```

## 2 Roman substitution cypher hacken

Met volgende tool kan je een tekst encypteren volgens een Roman substitution cypher:

<https://www.xarg.org/tools/caesar-cipher/>. De sleutel is dan hoeveel plaatsen er zijn opgeschoven. Als de sleutel "3" is, dan wil dat zeggen alle letters met 3 opschuiven (dus een a wordt een d) en andersom om te decrypteren (een d wordt een a).

Maak zelf een tool om een tekstfile (die je zelf kiest) de encypteren naar een nieuwe tekstfile. Geef de geëncrypteerde tekst aan aan medestudent, maar vertel die niet wat de sleutel is. Aan hem om de originele tekst opnieuw tevoorschijn te halen.

Tip: gebruik een dictionary met <char, double> op basis van

<http://www.gutenberg.org/files/5200/5200.txt> (lange Engelse tekst). Bij char zet je alle letters van het alfabet. Bij double hoeveel keer ze voorkomen in een grote tekst (% kans, dus <a,10> wil zeggen van alle karakters komt de a in 10% van de gevallen voor). Als je deze dictionary niet hebt uit vorige oefeningen, dan moet je die nog zelf maken. Zoek online zelf voor lange teksten op bijvoorbeeld wikipedia.

Tip: als <char, double> niet goed werkt, probeer dan <string, double> en zoek op woorden ipv individuele characters.

Dit werkt uiteraard enkel voor grote teksten.



## 3. Ontwerp een klasse CD

voor de opslag van de titel en de uitvoerder. De constructor maakt een nieuwe CD aan met de ingegeven titel en uitvoerder. Voeg twee methoden toe om de inhoud van de

velden titel en uitvoerder op te vragen. Voeg nog een veld toe waarin een lijst van liedjes staat. Zorg er ook voor dat je een methode hebt waarmee je een liedje kan toevoegen. Voeg een methode toe om alle informatie over de titel, uitvoerder en de lijst van liedjes omvormt naar tekst.

Maak een klasse Jukebox die een collectie van CD's heeft. Elke cd kan maar 1x voorkomen. Deze heeft een methode om cd's toe te voegen en om zichzelf naar file te schrijven.

Zorg ervoor dat je de CD met elk van hun nummers naar file kan schrijven en terug kan inladen in een nieuw JukeBox object via serialisation.

Maak in je Main een lijst van CD's. Maak een Shuffle methode die alle nummers van alle CD's oplijst in willekeurige volgorde. Voorzie een duidelijke weergave van de informatie in je console.

4. Maak een klasse ConsoleSettings hierin maak je variabelen voor

- i. Voorgrondkleur (evt als string)
- ii. Achtergrondkleur (evt als string)
- iii. Aantal opstartbeeps

Maak een klasse MultipleSettings, die een lijst bevat van consolesettings

- Zorg dat je een AddConsoleSettings methode hebt
- Deze klasse bevat een methode WriteToFile die het hele object naar file schrijft
- Deze klasse bevat een constructor die een filenaam meekrijgt en via de filenaam het object initialiseert. Gebruik een hulpmethode ReadFromFile

Maak in je main meerdere settings aan, zorg dat je ze kan opslaan in 1 multiplesettings object, en sla dit op in een file. Maak een tweede object aan die je creeert op basis van je file.

Maak een menu die je toelaat te vragen de hoeveelste setting je wil tonen en toon deze setting in tekstvorm in de console.

## Herhalingsoefening

- Maak een klasse ColorPez
  - Hierin heb je een Flavor enum (appel, citroen, aardbei, banaan)
  - Elke Pez heeft een smaak en een Consolekleur
  - Maak properties (1 readonly) en een constructor die alles invult
  - Voeg een ToString toe
- Maak een klasse PezPack
  - Hierin zit een queue van ColorPez objecten
  - Zorg dat je 2 constructors hebt
    - 1 die geen parameters heeft en 10 random ColorPez smaken toevoegt
    - 1 die een smaak en een kleur meekrijgt en 10 van deze objecten aanmaakt en toevoegt



- De ToString schrijft uit: hoeveel ColorPez erin zitten en de ToString van elke ColorPez
- Een Pick die het voorste element uit het pak eruit haalt en teruggeeft
- Maak een klasse PezDispenser
  - Deze heeft een Stack van ColorPez
  - Een eigen default constructor die alles leeg initialiseert
  - Een methode Fill die een PezPack meekrijgt en elke ColorPez uit het pack toevoegt (tot er max 20 inzitten)
    - Gebruik een while lus, blijf vullen tot er max 20 in de stack zitten en stop ook als je pack leeg is (Pick geeft null terug).
  - Een Pop die het bovenste element teruggeeft
- Maak in je main een pezDispenser en 2 PezPack objecten (1 random en 1 met zelf te kiezen smaak en kleur)
- Zorg dat je beide in de PezDispenser steekt
- Maak een menu
  - TakePez
  - PrintPez
  - SavePez
  - LoadPez
  - RefillPez die vraagt welke smaak en kleur en dit ineens toevoegt aan de dispenser
 Die alles uitvoert zoals verwacht