Algemene opmerkingen:

* Lees grondig de opgave en voer ze uit zoals vermeld. Een voorbeeld screenshot van de output vind je onderaan elk deel. Op het laatste blad vind je een volledig overzicht van de verwachte output.
* Het is niet omdat je 1 onderdeel niet volledig kan oplossen, dat de andere onderdelen niet zullen werken of je er geen punten voor krijgt. Zorg wel dat je onderdelen compileren/builden of iets simuleren zodat de rest van je opdracht wel werkt. Elk onderdeel apart moet wel minstens voor 2/3 in orde zijn om hierop punten te behalen, is dit niet het geval, krijg je voor dat onderdeel een 0.
* Schrijf altijd propere, gestructureerde, code, vermijd dubbele code en pas voor alle code, en variabele- en methodenamen de aangeleerde regels toe.

AFSLUITEN! Geef aan de docent aan dat je wil afsluiten en inleveren. Daarna sluit je je visual studio en maak je een zipfile met je familienaam.zip als naam, hierin zit je volledige map met je solution, project en examencode. PAS DAN MAG JE JE INTERNET INSCHAKELEN!!!

Upload via BB -> Programmeren 1 –> EXAMEN -> P1\_2017\_INHAAL

Veel success

# PROJECT/SOLUTIONOPSTELLING /1

1. Maak een nieuwe project en solution aan, met daarin een console applicatie project en geef als naam: je ACHTERNAAM. (bvb voor Pieter Jorissen zullen het project en de solution volgende naam hebben: Jorissen)
2. Schrijf in de Program.cs file, **boven** de Main methode de commentaar:

*Dit is het examen van* [JE VOORNAAM][JE ACHTERNAAM]

(dus voor pieter jorissen staat er in commentaar: Dit is het examen van Pieter Jorissen)

# DEEL 1 – Naam en diamant /7

A screenshot of a cell phone

Description generated with high confidence

SCREENSHOT DEEL 1: Het resultaat voor voorbeeld Lorenz Adriaensen en random getal 3.

In je Main methode programmeer je het volgende:

* 1. Een variabele die je voornaam bevat. Initialiseer die met **jouw** voornaam als waarde (dus bvb: Pieter)
  2. Een variabele die je achternaam bevat en initialiseer die met **jouw** achternaam als waarde (dus bvb: Jorissen)
  3. Schrijf naar de console uit (Je mag extra variabelen aanmaken) :

*Hallo [*VOLLENAAM], *je voornaam bestaat uit* [X] *letters, je achternaam uit* [Y].

1. Genereer nu een random getal, van 0 t.e.m. 9. Sla dit op in een variabele
2. Print nu in de console een diamantvorm af (zoals je in de bovenstaande screenshot kan zien) met het random getal dat je eerder aanmaakte als karakter. Dus stel dat je random getal 5 is, bestaat de diamant uit de cijfers 5.

# DEEL 2 – getallen en som zoeken /6

A close up of a logo

Description generated with very high confidence

SCREENSHOT DEEL 2: het resultaat

Je moet een programma schrijven dat de gehele getallen zoekt tussen 2 grenzen die deelbaar zijn door een bepaald getal. Ook bereken je de som van deze getallen.

In je Main methode programmeer je het volgende:

1. Maak volgende variabelen aan: de bovengrens, de ondergrens, de deler.
2. **Blijf** de gebruiker vragen naar de bovengrens, ondergrens en deler tot deze voldoen aan (bedenk zelf de code, je mag zelf extra variabelen toevoegen die je nodig denkt te hebben):
   1. **Alle** 3 de inputs zijn een geldig geheel getal.
   2. De bovengrens moet 2 groter zijn dan de ondergrens.
   3. De deler mag niet 0 zijn.

Zolang niet aan alle voorwaarden voldaan is herhaal je de vraagstelling dus. En geef je deze foutboodschap: “Geef geldige waarden in voor boven- ondergrens en deler”

Als alle 3 de inputs (bovengrens, ondergrens en deler) juist zijn, zoek je de **gehele getallen** tussen de onder- en bovengrens die deelbaar zijn door de ingevoerde deler. Hierna print je alle getallen van bovengrens tot ondergrens uit naar de console die deelbaar zijn door de deler (zoals in bovenstaande screenshot)

Ga de berekende waarden vooraf door volgend zinnetje: “Getallen tussen [BOVENGRENS] en [ONDERGRENS], deelbaar door [DELER]:”

Bereken ook de som van al deze getallen (dit mag in dezelfde lus berekend worden). Ga die vooraf door deze zin: “De som : [DESOM]”

# DEEL 3 – getallen /7

**In deze applicatie vraag je een gebruiker om een woord in te voeren en dan wat berekeningen uit op basis van het aantal letters in dit woord. Voor de naam Ivan wordt dit:**

1. Vraag aan de gebruiker om een woord in te geven en sla die op een variabele met de naam inputString.
2. Ga na hoeveel letters er in die naam zitten en sla dit op in een variabele met de naam inputLength.
3. Als inputLength deelbaar is door 4 maar niet door 5, dan schrijf je naar het scherm de zin “Number of characters in name is divisible by 4 but not by 5”. Test dit door het woord “Jack” in te geven.
4. Indien aan de dubbele voorwaarde hierboven niet is voldaan test dan of het woord gelijk is aan naam “Hello” is ingegeven. Indien dit het geval is, dan schrijf je “World” naar het scherm.
5. Indien een ander woord is ingegeven, dan schrijf je “What a nice word “ gevolgd door het ingegeven woord, naar het scherm.
6. Daarna schrijf je in elk geval “Number of characters in word “ gevolgd door inputLength, naar het scherm.
7. Tot slot schrijft de applicatie evenveel ‘=’ naar het scherm, als er letters in het ingegeven woord zitten. Tussen elk ‘=’ staan evenveel spaties, als de plaats van de letter in de naam.

Dus na de eerste letter 1 spatie.

Na de tweede letter, 2 spaties.

Na de derde letter, 3 spaties.

Na de vierde letter, 4 spaties.

…

Tip: gebruik een geneste for lus.