

Naam student in groep:	Klas:
STUDENT 1:	
STUDENT 2:	
STUDENT 3:	

Dit is de peilingslijst voor het project "Torens van Hanoi". Laat de punten op tijd aftekenen en **kijk goed naar de beoordelingscriteria** om te weten wat er precies van je verwacht wordt.

Kerntaak	Werkproces	Op te leveren product(en)	Niveau	Paraaf docent
KT1	B1-K1-W1 Stelt de opdracht vast	1. Wireframes schetsen applicatie	Starter	
	B1-K1-W2 Levert een bijdrage aan het projectplan	2. Huisstijl voor eigen documenten	Starter	
		3. Samenwerkingscontract	Starter	
		4. Bereikbaarheidslijst	Starter	
		5. Taakverdeling	Starter	
		6. Logboek aanmaken	Starter	
	B1-K1-W4 Bereidt de realisatie voor	7. Materiaal en middelenlijst	Starter	
<b>Handtekening Kerntaak 1</b> (Ga ondertussen verder met de volgende punten):				
KT2	B1-K2-W1 Realiseert (onderdelen van) een product	8. Diagram: uitrekenen m.b.v. For-loop	Starter	
		9. Algoritme in C#: uitrekenen m.b.v. een For-loop	Starter	
		10. Algoritme in C#: uitrekenen m.b.v. machtsverheffen	Starter	
		11. Diagram: uitrekenen m.b.v. recursie	Starter	
		12. Algoritme in C#: uitrekenen m.b.v. recursie	Starter	
		13. Applicatie laat tijdsberekening zien	Starter	
		14. Applicatie laat uitvoer netjes zien	Starter	

		15. Vragenlijst Torens van Hanoi ingevuld	Starter	
KT2	B1-K2-W2 Test het ontwikkelde product	16. Technische test For-loop	Starter	
		17. Technische test Machtsverheffen	Starter	
		18. Technische test recursie	Starter	
Handtekening Kerntaak 2 (Ga ondertussen verder met de volgende punten):				
KT3	B1-K3-W1 Optimaliseert het product	19. Eventuele aanpassing aan de hand van de technische testen	Starter	
	B1-K3-W2 Levert het product op	20. Volledige project documentatie en 3 werkende onderdelen binnen een enkele solution	Starter	
Opmerkingen:				Naam docent:
				Handtekening docent:

Op te leveren product(en)	Eisen
1. Wireframes	Wireframes van alle schermen binnen jullie applicatie. Je gaat als groep nadenken over de inrichting van de applicatie en probeert het voor de eindgebruiker zo makkelijk mogelijk te maken de uitkomst te zien bij een ingevoerd aantal schijven.
2. Huisstijl voor eigen documenten	<b>Een professioneel ogende huisstijl</b> met voor ieder document: <ul style="list-style-type: none"> <li>- Een voorblad met informatie over groep (namen en klassen), de naam van het project, datum en de titel van het document</li> <li>- Een simpel zelfgemaakt logo op het voorblad en aan de bovenkant van iedere pagina</li> <li>- Paginanummering</li> <li>- Inhoudsopgave automatisch gegenereerd met Word</li> </ul>
3. Samenwerkingscontract (In huisstijl)	<b>Regels opgesteld wat betreft de samenwerking binnen de groep:</b> <ul style="list-style-type: none"> <li>- Tijdsgebonden en concrete afspraken [google: SMART]</li> <li>- Afspraken over melden bij afwezigheid en te laat komen</li> <li>- Afspraken over vergaderingen</li> <li>- Afspraken over delen documenten (welke cloud-storage en afspraken over bestandsnamen/locaties)</li> <li>- Afspraken over bijdragen aan het project</li> <li>- Consequenties bij niet houden aan afspraken</li> </ul>
4. Bereikbaarheidslijst	<b>Contactgegevens van alle groepsleden</b> (zodat jullie elkaar kunnen bereiken)
5. Taakverdeling (In huisstijl)	<b>Een overzicht</b> van wie, wat, wanneer gaat doen en hoe lang dat ongeveer gaat duren.
6. Logboek aanmaken	<b>Een document waarmee groepsleden hun bijdragen aan het project kunnen bijhouden.</b> <ul style="list-style-type: none"> <li>- Wanneer heb je wat gedaan?</li> <li>- Hoe lang duurde dat?</li> <li>- Wat waren bijzonderheden (problemen waar je tegen aanliep)?</li> </ul>
7. Materiaal- en middelenlijst (In huisstijl)	<b>Een lijst met materialen en middelen die nodig zijn tijdens dit project.</b> Denk aan versienummers bij software en geef korte beschrijvingen waar de materialen en middelen voor gebruikt gaan worden.

	Er moeten ook minimale hardware-eisen opgesteld worden.
8. Diagram: uitrekenen m.b.v. For-loop	Een diagram waarop je visueel uitbeeldt hoe het algoritme gaat werken. <b>Dit moet een flowchart of PSD (Programmastructuur-diagram) zijn gemaakt in een tool bedoeld voor het tekenen van diagrammen (bijv.: Visio, Draw.io of Edraw).</b>
9. Algoritme in C#: uitrekenen m.b.v. een For-loop	Een in C# gecodeerd algoritme waarmee uitgerekend kan worden hoeveel zetten het kost om alle schijven te verplaatsen, bij een door de gebruiker ingevoerd aantal schijven. <b>Er wordt gebruik gemaakt van een for-loop voor de berekening.</b> In de uitvoer van het programma mag het getal niet als 'E' (wetenschappelijke notatie) weergegeven worden.
10. Algoritme in C#: uitrekenen m.b.v. machtsverheffen.	Een in C# gecodeerd algoritme waarmee uitgerekend kan worden hoeveel zetten het kost om alle schijven te verplaatsen, bij een door de gebruiker ingevoerd aantal schijven. <b>Er wordt gebruik gemaakt van machtsverheffing voor het uitrekenen.</b> In de uitvoer van het programma mag het getal niet als 'E' (wetenschappelijke notatie) weergegeven worden.
11. Diagram: uitrekenen m.b.v. recursie	Zie 8, maar dan voor recursie.
12. Algoritme in C#: uitrekenen m.b.v. recursie	Een in C# gecodeerd algoritme waarmee uitgerekend kan worden hoeveel zetten het kost om alle schijven te verplaatsen, bij een door de gebruiker ingevoerd aantal schijven. <b>Er wordt gebruik gemaakt van recursie. Er mag geen for-loop of machtsverheffing aanwezig zijn binnen dit algoritme.</b> In de uitvoer van het programma mag het getal niet als 'E' (wetenschappelijke notatie) weergegeven worden.
13. Applicatie laat tijdsberekening zien	Na het berekenen van het aantal zetten met een van de algoritmes dient de applicatie te laten zien hoe lang het duurt de schijven te verplaatsen. <b>We gaan er vanuit dat een zet een seconde kost.</b> Laat t/m eeuwen zien hoe lang het duurt de schijven te verplaatsen.
14. Applicatie laat uitvoer netjes zien	<b>De applicatie oogt als een professionele applicatie.</b> Er is aandacht voor witruimte binnen de applicatie en de knoppen zijn a.d.h.v. een verklaarbare logica geplaatst.
15. Vragenlijst Torens van Hanoi ingevuld	<b>De vragenlijst bij dit project is beantwoord.</b> En ieder groepslid is op de hoogte van de vragen en antwoorden in dit document. <i>De docent stelt ieder groepslid een willekeurige vraag uit dit document.</i>
16. Technische test For-loop (In huisstijl)	Een test met ten minste 10 vragen over de technische werking van het for-loop algoritme. Er is aandacht voor controle van gebruikersinvoer. Vaak is het zo dat als bij een testvraag 'Ja' wordt geantwoord is de test geslaagd (Zorg er dus voor dat een tester bij alle positieve uitkomsten 'Ja' moet omcirkelen.)

	<p><b>Het doel is testen of de applicatie in geen gevallen kan crashen of onjuist kan reageren richting de gebruiker.</b></p> <p><i>Er is bewijs op papier dat ten minste 2 klasgenoten deze test hebben uitgevoerd. Niet geslaagde testen moeten opnieuw worden uitgevoerd na het verbeteren van de applicatie.</i></p> <p>Let er op dat de test ook algemene informatie bevat, zoals Onderwerp, Datum, Tester etc. Maak gebruik van je huisstijl sjabloon.</p>
17. Technische test machtsverheffen (In huisstijl)	Zie 16, maar dan voor het machtsverheffen-algoritme.
18. Technische test recursie (In huisstijl)	Zie 16, maar dan voor het recursie-algoritme.
19. Eventuele aanpassing aan de hand van technische testen	<p>Bewijs in de logboeken over aanpassingen die gemaakt zijn na de technische tests. Je schrijft een kort testverslag waarin je aangeeft:</p> <ul style="list-style-type: none"><li>- Wie heeft de testen gemaakt, wanneer en op wat voor apparaat?</li><li>- Wat voor bijzonderheden kwamen er naar boven tijdens het testen?</li><li>- Zijn de testen geslaagd? Waarom niet?</li><li>- Wat gaat er gedaan worden om de testen te laten slagen?</li><li>- Zijn de nieuwe testen geslaagd?</li><li>- Wat zijn jullie conclusies na het testen?</li></ul>
20. Volledige projectdocumentatie en 3 werkende onderdelen binnen 1 solution	Er is een mappenstructuur met alle documentatie, de werkende onderdelen in 1 solution (source-code <u>en</u> het .exe-bestand van de applicatie)