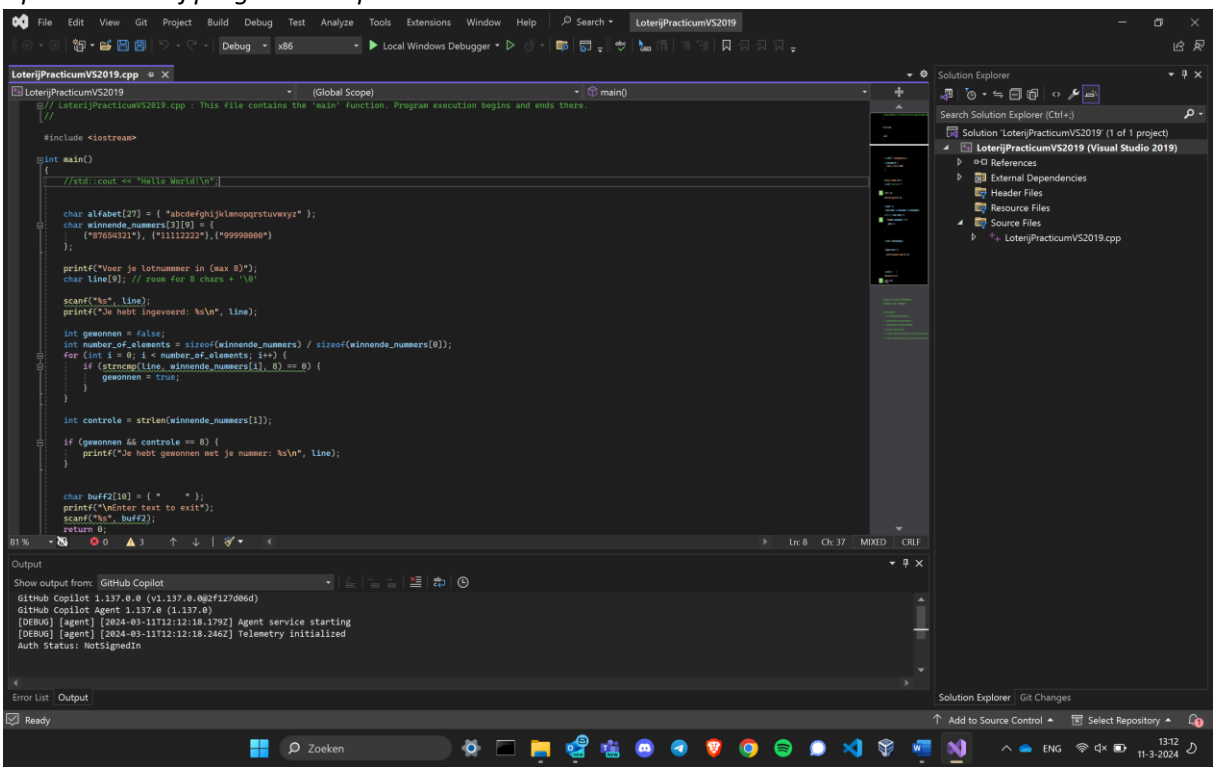
CST1 - Practicum les 5 – buffer en integer overflow

Opdracht

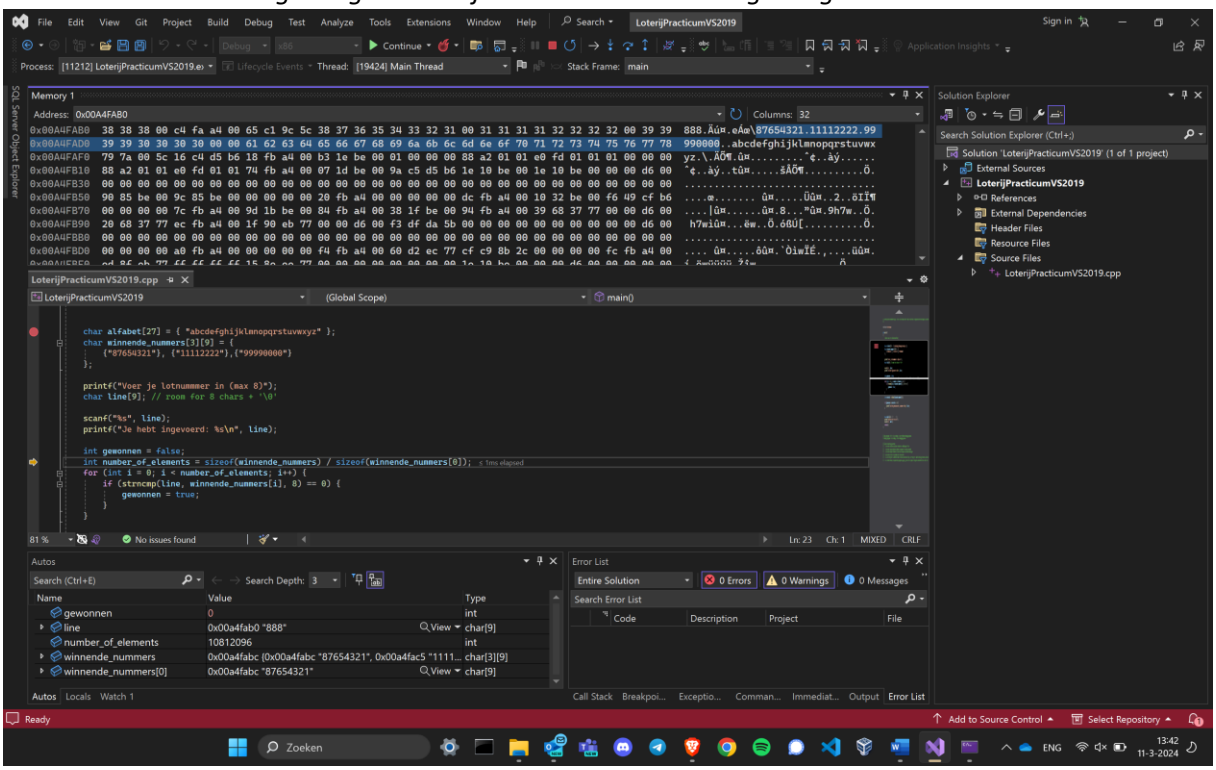
1. *Open het loterij programma op blackboard in Visual Studio.*



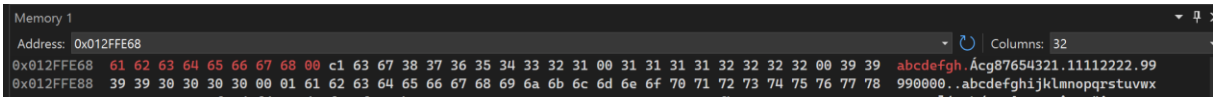
2. *Lees de code en beschrijf wat dit programma doet. NB: hiervoor moet je elke regel code begrijpen*

   Het programma begint met een array van alle letters van het alfabet. Vervolgedns worden de winnende nummers opgeslagen in een array.
   Het programma vraagt de gebruiker om een invoer. Vervolgens komt er een array genaamd line dat een capaciteit van 9 heeft. 8 hiervan zijn de gebruikersinvoer en 1 is de null-terminator.
   Scanf leest de invoer uit de variabele line. Vervolgens wordt er met printf een bepaalde tekst afgedrukt met daarachter de variabele line.
   Vervolgens wordt gewonnen op false gezet. Er is een for loop die wordt gebruikt om te kijken of de ingevoerde tekst overeenkomt met een van de winnende nummers. Als deze overeenkomt wordt gewonnen op true gezet.
   Controle krijgt de lengte van het tweede winnende nummer. Tweede aangezien een array begint bij 0 en er hier 1 staat.
   Als de invoer overeenkomt met een van de winnende nummers en de lengte van het tweede nummer 8 lang is, wordt er op het scherm afgedrukt dat de gebruiker heeft gewonnen gepaard met het nummer.
   Er wordt om een invoer gevraagd om het programma te sluiten.
   Vervolgens wordt er aangegeven dat het programma correct is afgerond dmv return 0.

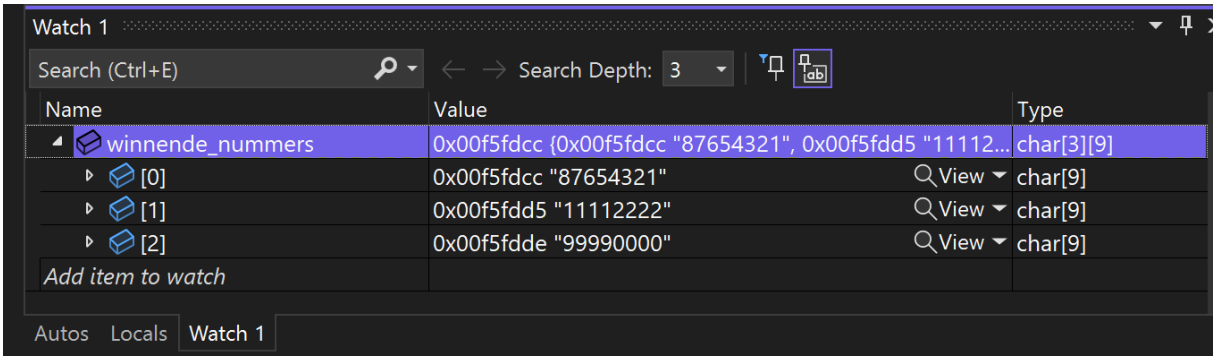
3. *Toon mbv een tool het geheugen zodat je kan zien waar in het geheugen de variabele zitten*



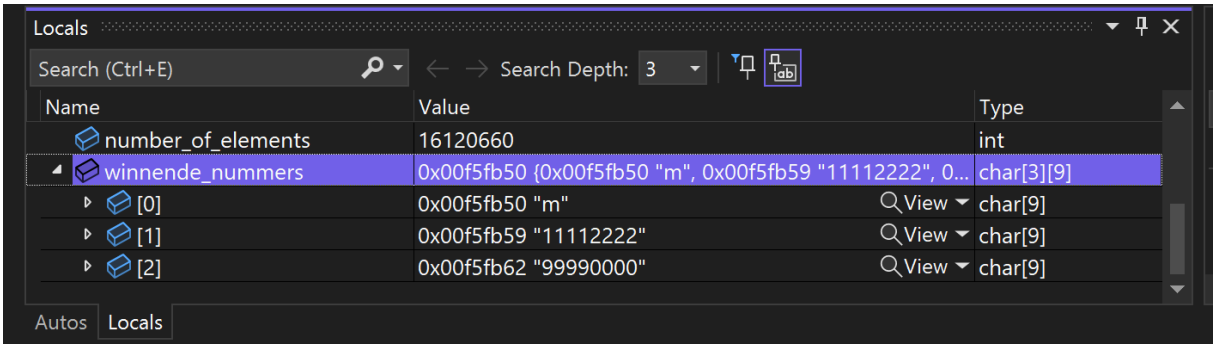
4. *Laat zien bij welke invoer er een bufferoverflow plaats vindt en welke variabele er deels overschreven wordt. Leg uit bij welke lengte van de invoer dit gebeurt.*
We kunnen onze invoer zien in het memory window. De invoer van 8 bytes is terug te zien. Tussen de invoer en de eerste waarde van het winnende nummer zit zo te zien 4 bytes. Dit zie ik aan het stukje ".Ácg". De padding is dus 4 bytes.



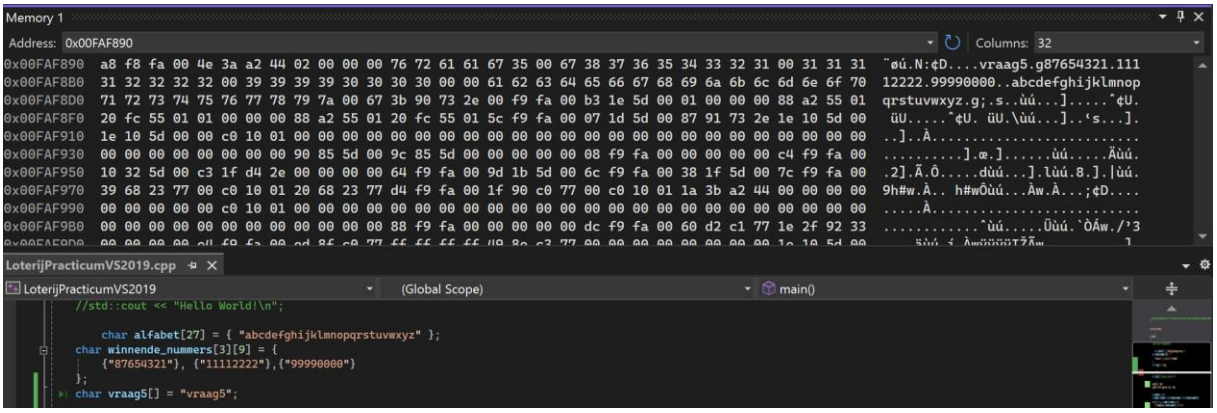
We kunnen de variabelen terug zien onder locals. Hierbij kunnen wij de waarde van de winnende nummers terug zien.



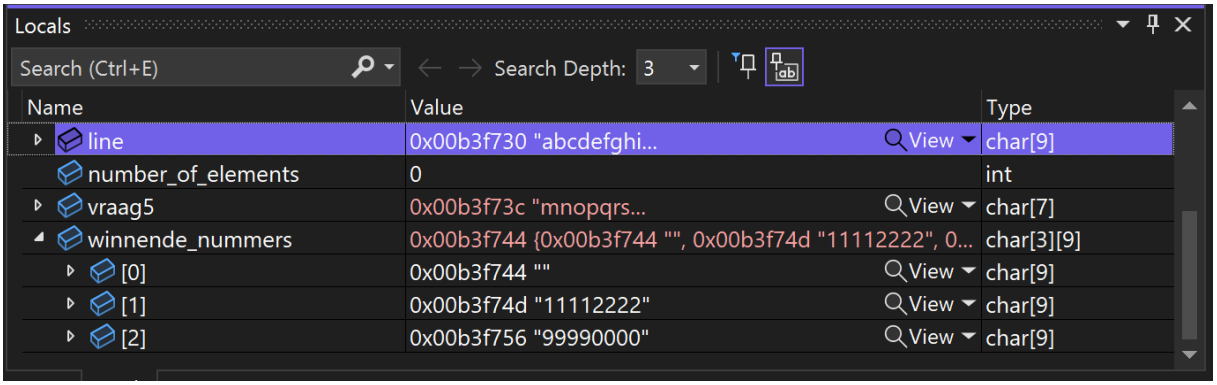
Om de invoer te bepalen heb ik de invoer van 8 bytes opgeteld met de padding 4 bytes met vervolgens 1 extra byte om de waarde te overschrijven. Zodra ik “abcdefghijklm” invoer, kun je zien dat de waarde van het eerste winnende nummer wordt overschreven met “m”. Dit is een invoer van 13 bytes.



5. *Declareer een extra variabele in de code waar de lengte van de vorige vraag anders is. Laat dit zien mbv het memory window. Leg ook uit hoe je op basis van je code de nieuwe lengte van de overflow had kunnen voorspellen.*
Ik heb een variabele aangemaakt “vraag5”. Deze heeft een lengte van 6 karakters. Dit betekend dat er 6 bytes bij zijn gekomen + 1 voor de nullterminator. Hierdoor krijg je 13+6+1= 20 bytes.

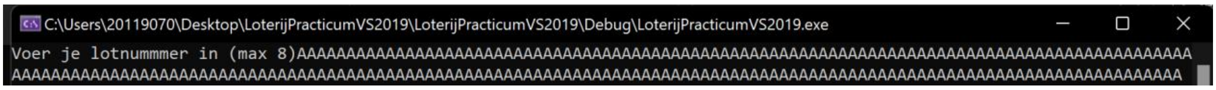


Bij een invoer van 20 bytes wordt het eerste winnende nummer aangepast.

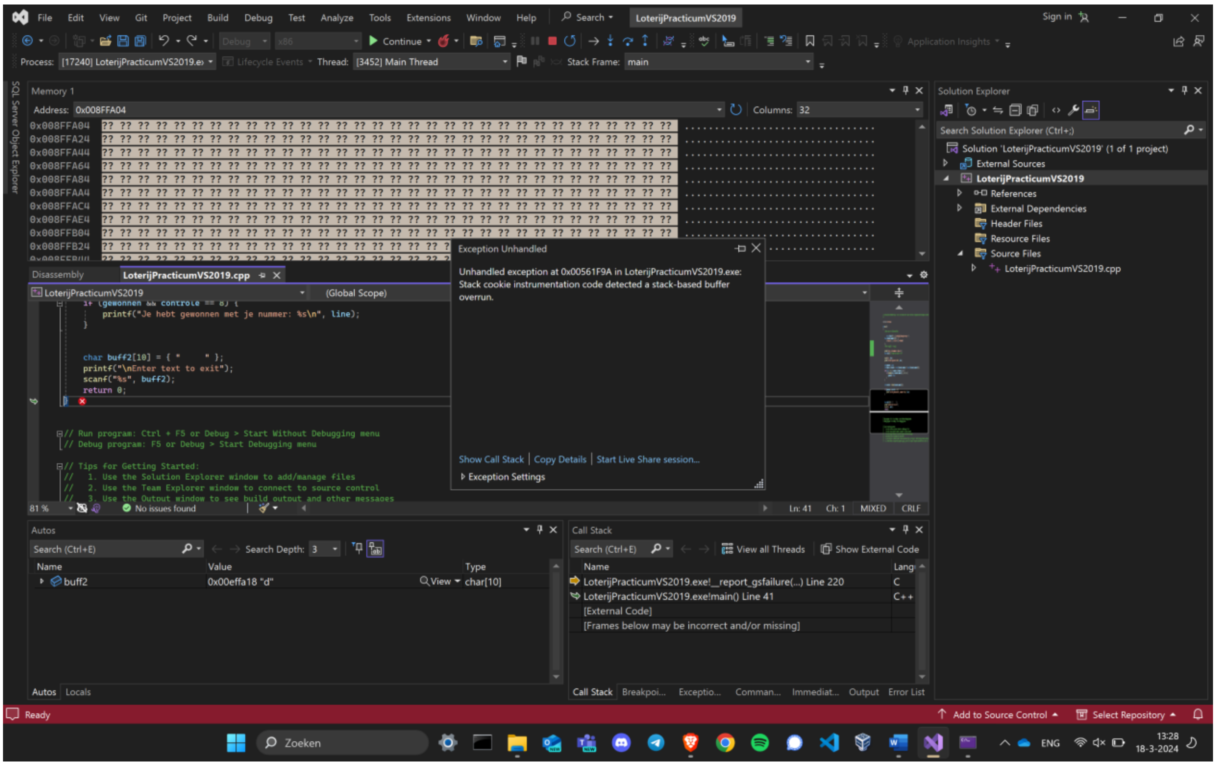


6. *Laat zien dat een lange invoer de stack cookie beveiliging (of wel canary genoemd) het programma stopt.*

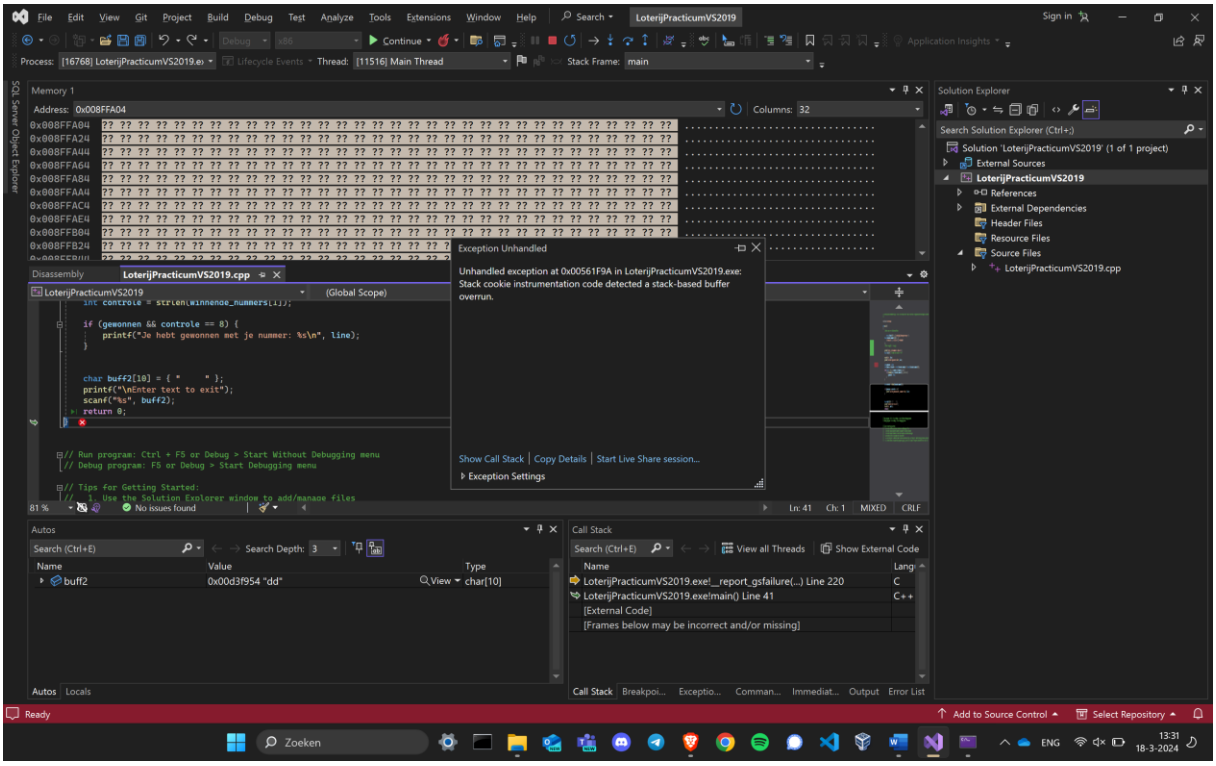
Om de canary af te laten gaan geef ik het programma een lange invoer.



Na de lange invoer krijg je een exception error te zien. Dit geeft aan dat er een acces violation is voor het pogen te schrijven op een bepaalde locatie van het geheugen. Dit is de canary die het schrijven detecteert en daarom het programma stopt.

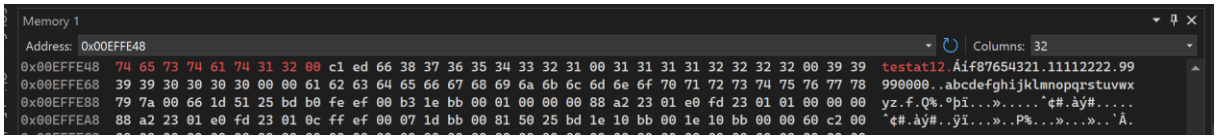


7. *Laat zien met breakpoints wanneer het programma stopt (welke instructies na de bufferoverflow worden nog wel uitgevoerd en welke niet meer)*
   Alles voor de return wordt nog uitgevoerd. Als je na de return 1 verder stapt, krijg je een execption error.

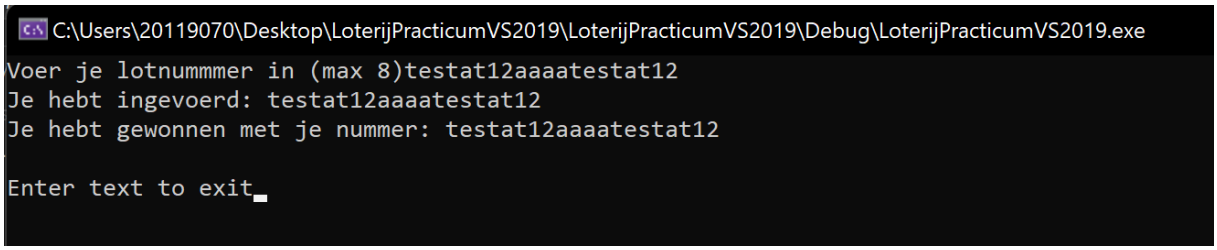


8. *Probeer te winnen met een ongeldig (te lang) lotnummer*

Testat12 als invoer is terug te zien in het geheugen.



De eerste 8 bytes van de invoer is het winnende nummer dat wij gebruiken voor de invoer. De 4 bytes erna zijn voor de padding. De 8 bytes daarna zijn het eerste winnende nummer. Dit kan worden gebruikt voor het overschrijven van de winnende nummer in de array. Door je invoer + 4 bytes + weer je invoer te typen, kun je jouw invoer in de array plaatsen. Hierdoor heb ik mijn ingevoerde nummer zowel als invoer als in de lijst staan van de winnende nummers. Hierdoor kan je winnen met een ongeldig nummer.



9. *Laat zien dat met minimale aanpassing van de code te lange invoer niet meer mogelijk is. Zie: https://www.tutorialspoint.com/c_standard_library/c_function_scanf.htm*
   Aan de website te zien moet de line “scanf("%s", line);” aangepast worden naar “scanf("%8s", line);”. Je specificeert hiermee hoe groot je de input verwacht.

10. Je kan ook een langer altijd winnend lotnummer vinden dan bij vraag 8 vinden. Vindt dit nummer en geef een uitleg met het memory window. NB (vergeet niet de aanpassing bij vraag 9 ongedaan te maken)
    Bij vraag 10 kun je ongeveer hetzelfde doen als vraag 8. Om een langer nummer te krijgen dat wint, kunnen we de eerste 2 nummers aanpassen ipv alleen de eerste. Hierbij krijg je het volgende: input (8 bytes) + 4 bytes padding + input + 1 byte padding (door het overschrijven verdwijnt de nullterminator dus we voegen een byte toe) + input. Dit overschrijft de eerste 2 nummers met hetzelfde als de input waardoor je een langer winnend nummer hebt dan bij vraag 8. Input: “testat12AAAAtestat12Atestat12”.