Questions

# Questions: arrays1

Lav et program, som indeholder et array Numbers med værdierne {2,3,5,7}. Print disse værdier ud.

**Fremgangsmåde:**

* 1. lav og åben filen arrays1.c
  2. opret et array som indeholder værdierne {2,3,5,7}
     1. Hint: Hvis man skal bruge et heltal, hvilken datatype er så passende?
  3. Lav et loop, som printer alle værdier af arrayet ud en efter en.
     + 1. Hint: Arrays er indexeret fra 0 til N-1
     1. > Numbers[0] indeholder værdien 2
  4. Compile og kør programmet fra VS Code
  5. Compile og kør programmet fra terminalen

# Questions: arrays2 Kompile og Kør koden, se arrays2.c, nogle gange fra terminalen:

**Fremgangsmåde:**

1. Compile arrays2.c

gcc arrays2.c

1. Kør programmet fra terminalen

./a.exe

./a.exe

./a.exe

1. Hvad gør koden, og hvad gør dette til dynamisk allokering af hukommelse?
2. Er dette heap eller stack allokering?
3. Hvad er fordele ved hhv. heap og stack allokering

# Questions: pointers

Lav et program som indeholder en int variable num, og en int pointer p\_num. programmet skal gemme hukommelsesaddressen af num i p\_num. Programmet skal også bruge p\_num til at ændre værdien af num.

**Fremgangsmåde:**

1. Lav programmet pointers1.c, som har følgende variable
   1. int num
   2. int\* p\_num
2. Sæt p\_num til num-hukommelse-addressen.
   1. hint: Hvordan får man addressen af en variable (reference)
3. Ændre værdien af num, ved at bruge p\_num
   1. hint: hvordan de-referencer man en pointer variabel? (og hvad betyder det)
4. print værdien og addressen af num og p\_num
5. kør programmet

# Questions: descriptive1

Lav en program, som kan printe summen af følgende tal ud:

{151,123,7,1,-31,0,17,9921,9,15,-1003,-3,17}

**Fremgangsmåde:**

1. Definer et array arr som indeholder alle talene ovenover.
2. Definer en variabel sum, og sæt den til 0.
3. vha et loop summer alle tallene i arr i via variablen sum.
4. Print sum ud.
5. Compile og kør programmet

# Questions: descriptive2

Lav en program, som kan regne middelværdien af følgende tal: {151,123,7,1,-31,0,17,9921,9,15,-1003,-3,17}.

Gennemsnit:

**Fremgangsmåde:**

1. Tag summen af alle de tal brugeren har taget som input
   * 1. brug sidste opgave som udgangspunkt
2. gem gennemsnittet i en variabel gennemsnit, som er defineret som **sum / N** ud.
3. Compiler og kør programmet fra terminalen og vs code

# Questions: descriptive3

Lav en program, som kan regne middelværdien og standard afvigelsen af alder (og hvis du får tid: højde, og vægt) på en liste af patienter i en patient journalen. Overvej hvilke libraries vi skal bruge (Hint: Hvordan man udregner og i c. Findes der muligvis et library med math-funktioner?)

gennemsnit:

standard afvigelse:

1. definer et 2D float array measurements med størrelsen 3x10, og 2 1D arrays af af længden 3, som hedder hhv gennemsnit og afvigelse
2. Definer et 1D array measurement\_names, som indeholder 3 strings: "Age", "Height", "Weight"
3. Udregn gennemsnittet af alder, højde og vægt, på patienter med værdierne til højre

Hint: Det har i lige gjort

1. Udregn standard afvigelse af alder, højde og vægt (skip dette step indtil sidst)
2. Print descriptive tabel ud i terminalen (mean std)**)** for alder, højde og vægt,
3. Compile og kør programmet

N\_patients = 10

N\_measurements = 3

Alder: 23., 35., 67., 42., 54., 18., 29., 72., 31., 88.

Højde: 1.78, 1.62, 1.50, 1.90, 1.70, 1.55, 1.82, 1.60, 1.75, 1.85

Vægt:70.5, 55.3, 60.2, 85.7, 72.4, 50.1, 68.9, 64.0, 75.3, 90.2

Answers

# Answer: arrays1

Se arrays1\_sol.c

# Answer: arrays2

Se arrays2\_sol.c

# Answer: arrays2

1. dynamic\_array\_size = 1 + rand() % Max\_array\_size;

Denne giver en dynamisk allokering af data, som betyder at størrelsen af arrayet er anderledes hver gang man kører programmmet

1. når man bruger dynamisk hukommelse er det altid heap
2. heap gør at man altid kan justerer hukommelses størrelsen.

Men det kræver også at vi selv holder styr på størrelsen af arrays,

Samtidig er det også vigtigt at man frigiver hukommelsen når man er færdig med det.

# Answer: pointers

Se pointers1\_sol.c

# Answer: descriptive1

Se descriptive1\_solution.c

# Answer: descriptive2

Se descriptive2\_solution.c

# Answer: descriptive3

Se descriptive3\_solution.c