

SLUTTEKSAMEN IBE102 WEBUTVIKLING HØGSKOLEN I MOLDE VÅREN 2021

Oppgaven teller 100 % av sluttarakter og gis bokstavkarakter A-F.

Alle hjelpemidler tillatt (bøker, notater, PC, programvare).

Tips: Unngå æøå i koden hvis du bruker få problemer med dette.

Samarbeid med andre personer under eksamen er ikke tillatt. F.eks. telefon, epost, SMS, Discord, messenger, video, zoom, diskusjonsforum. Uansett form.

Alle svar plagiatsjekkes av programvare. Mistanke om ulovlig samarbeid vil etterforskes. Det *kan* medføre at avdelingen oppretter fuskesak og annullering eller utestengelse.

<https://www.himolde.no/studier/eksamen/regler-fusk-og-plagiat/>

Oppgavene har lik vekt. Svar på så mye du greier. Skriv kommentar i koden om det du velger å ikke besvare.

Du skal levere en ZIP-fil med en mappe "kand012" hvis du er kandidat 12. På den mappen skal ligge følgende filer:

verdier.php (svar til oppgave 1)

teksthusker.php (svar til oppgave 2)

nyttpunkt.php (svar til oppgave 3)

hjelp.php (svar til oppgave 3)

tusentall.php (svar til oppgave 4)

tusentall2.php (svar til oppgave 4)

Sensor vil pakke ut koden og forsøke å kjøre disse via localhost. I oppgave 3 bruker koden en SQL-database.

Koden skal være ryddig skrevet med innrykk. VSCode og noen andre kode-editorer vil rykke inn korrekt automatisk, men du kan også bruke annet (som <http://www.phpformatter.com/>).

Grums skal fjernes før levering. Med "grums" menes data og kode som ikke er i bruk.

Merk koden i kommentar, med filens navn og ditt kandidatnummer, for sikkerhets skyld.

Oppgave 1: Verdivising (25 %)

Lag et program "verdi.php" som kan regne med heltall og vise resultater grafisk.

Programmet skal ha følgende skall:

```
<?php // verdi.php, kandidat ...  
$verdier = array(3, 1, 14, 8, 19, 5, 14);  
...  
?>
```

Du kan anta at alle tallene er heltall. Med verdiene over, vil resultatet bli som i neste skjermbilde:



```
← → ↻ ⓘ localhost/IBE102/eksv21/verdi.php  
Verdiene er: 3, 1, 14, 8, 19, 5, 14,  
Snitt: 9.1428571428571  
  
Verdiene vises sortert, og med grønn farge for de som er under snittet:  
1, 3, 5, 8, 14, 14, 19,  
  
Hver verdi er vist med antall stjerner:  
***(3)  
*(1)  
***** (14)  
***** (8)  
***** (19)  
***** (5)  
***** (14)  
  
Stjernene er grønne for den del som er under snittet:  
***(3)  
*(1)  
***** (14)  
***** (8)  
***** (19)  
***** (5)  
***** (14)
```

Det du ser på forrige skjermbilde er altså fire avsnitt (P) som dette programmet skal generere.

De 4 avsnittene inneholder følgende:

1. (Øverst) Verdiene vises i samme rekkefølge som de står i arrayet. På neste linje står gjennomsnittet av verdiene. Det er ditt program som skal regne ut og vise dette snittet. I eksemplet er snittet 9.143.

2. En sortert liste med verdiene blir vist. Hver verdi skrives ut med grønn farge hvis den er under snittet og rød farge hvis den er over snittet. Derfor blir 8 vist som grønn (fordi $8 < 9.143$) mens 14 vises som rødt.
3. Hver verdi vises på egen linje, først med et antall stjerner (*) likt verdien, slik at verdien 3 vises med `***(3)`
4. (Nederst) Samme som over, men hver stjerne under snittet farges grønn, mens de over farges rød. For verdien 3 er alle grønne, da 3 er under snittet. For verdien 14 (nederst) blir de 9 første grønne, fordi de representerer den delen under snittet på 9.143.

Hvis jeg skriver inn andre verdier får jeg (selvsagt) et annet bilde som resultat:

```
← → ↻ ⓘ localhost/IBE102/eksv21/verdi.php

Verdiene er: 3, 4, 5, 6, 7, 8,
Snitt: 5.5

Verdiene vises sortert, og med grønn farge for de som er under snittet:
3, 4, 5, 6, 7, 8,

Hver verdi er vist med antall stjerner:
***(3)
****(4)
***** (5)
***** (6)
***** (7)
***** (8)

Stjernene er grønne for den del som er under snittet:
***(3)
****(4)
***** (5)
***** (6)
***** (7)
***** (8)
```

Oppgave 2: Teksthusker (25 %)

Du skal lage programmet “teksthusker.php”. Den skal bruke cookie til å huske tekst i 1 time. Her ser du skjermbildet for en bruker som aldri har vært på siden før:

The screenshot shows a web browser window with the address bar displaying `localhost/IBE102/eksv21/teksthusker.php`. The page contains a form with the following elements:

- A text box labeled "Forrige tekst:" containing the text "(sist endret: Ikke vært endret før)".
- A text box labeled "+tekst som skal legges til:".
- A text box labeled "=Ny tekst:".
- A label "Skriv inn tekst:" followed by an empty text input field.
- A label "Slett tekst:" followed by an unchecked checkbox.
- A button labeled "Utfør".

Øverst er en tabell med 3 rader og 2 kolonner, der høyre kolonne skal vise tekst, men er tom på bildet over.

Hvis bruker skriver inn en tekst og trykker “Utfør” skal skjemaet sendes inn. Tilbake vil vi få et oppdatert skjema som viser ny tilstand, med fortsatt mulighet for å fylle ut ny tekst:

The screenshot shows the same web browser window, but the address bar now includes a query parameter: `localhost/IBE102/eksv21/teksthusker.php?nytekst=Jeg`. The form elements are updated as follows:

- The "Forrige tekst:" text box now contains "(sist endret: 18/05-21 kl 14:06:30)".
- The "+tekst som skal legges til:" text box now contains "Jeg".
- The "=Ny tekst:" text box now contains "Jeg".
- The "Skriv inn tekst:" label and input field remain the same.
- The "Slett tekst:" checkbox remains unchecked.
- The "Utfør" button remains the same.

Skriver man videre inn ny tekst og trykker “Utfør”, skjer det samme, og tilstanden endres, ved at den ny teksten føyes til den gamle, som vist i neste skjermbilde:

← → ↻ ⓘ localhost/IBE102/eksv21/teksthusker.php?nytekst=+holder+ditt+hode

Forrige tekst: (sist endret: 18/05-21 kl 14:08:13)	Jeg
+tekst som skal legges til:	holder ditt hode
=Ny tekst:	Jeg holder ditt hode

Skriv inn tekst:

Slett tekst: ☐

Hvis brukeren markerer sletting og trykker “Utfør”, som under:

Slett tekst: ☒

skal programmet sørge for at teksten blir blanket ut og vi får se følgende skjermbilde:

← → ↻ ⓘ localhost/IBE102/eksv21/teksthusker.php?nytekst=&slettalltekst=on

Alle tekst slettet!

Forrige tekst: (sist endret: 18/05-21 kl 14:09:00)	
+tekst som skal legges til:	
=Ny tekst:	

Skriv inn tekst:

Slett tekst: ☐

Bruker kan også krysse av for “Slett tekst” og skrive inn ny tekst samtidig, som her:

← → ↻ ⓘ localhost/IBE102/eksv21/teksthusker.php?nytekst=Nei%2C+og+2%2B2%3D5&slettalltekst=on

Alle tekst slettet!

Forrige tekst: (sist endret: 18/05-21 kl 14:19:53)	
+tekst som skal legges til:	Nei, og 2+2=5
=Ny tekst:	Nei, og 2+2=5

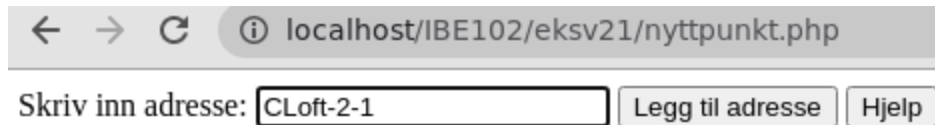
Skriv inn tekst:

Slett tekst: ☐

Programmet skal bruke samme variabelnavn som vist i skjermbildene. Dato skal vises nøyaktig som i skjermbildene. Hvis bruker skriver inn HTML skal teksten ufarliggjøres før den lagres. Skriv (i en kommentar) hvor du tror teksten blir lagret.

Oppgave 3: Adressevalidering og registrering (25 %)

I forbindelse med overvåkning av det trådløse nettet, skal vi lage en side der man kan legge inn tilkoblingspunkt, som så lagres i databasen “byggdata”.



← → ↻ ⓘ localhost/IBE102/eksv21/nyttpunkt.php

Skriv inn adresse:

Programmet “nyttpunkt.php” lar brukeren skrive inn koden for et tilkoblingspunkt i et tekstfelt. Når “Legg til adresse” er trykket, vil samme programmet ta imot koden, vaske koden, og validere den (se lenger ned i oppgaven), og legge den til i en tabell “opunkt”, hvis den ikke er der fra før.

Når alt gikk greit, skal skjermbildet se slik ut:



← → ↻ ⓘ localhost/IBE102/eksv21/nyttpunkt.php?adr=CLOft-2-1

Adressen [CLOft-2-1] ble lagt inn

Skriv inn adresse:

Hvis adressen ikke ble godkjent, som i neste bilde, får man beskjed om det:

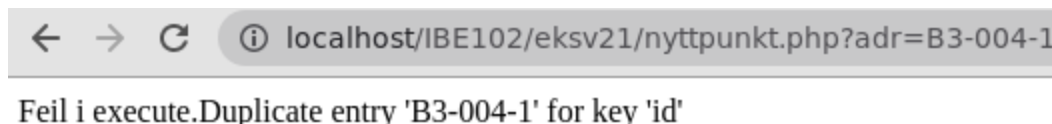


← → ↻ ⓘ localhost/IBE102/eksv21/nyttpunkt.php?adr=AKjel-04-6

Adressen [AKjel-04-6] ble ikke godkjent

Skriv inn adresse:

Hvis adressen var opptatt, eller noe annet gikk galt, skal en feilmelding komme, som her:



← → ↻ ⓘ localhost/IBE102/eksv21/nyttpunkt.php?adr=B3-004-1

Feil i execute.Duplicate entry 'B3-004-1' for key 'id'

Når knappen for “Hjelp” trykkes, vises følgende hjelpeprogram i et eget vindu (se vedlagte hjelp.php):

Et tilkoblingspunkt er identifisert med en spesiell kode som viser bygning, etasje, romnummer og pluggnummer.

- Bygningskodene er A, B eller C.
- Etasjene er Kj (for kjeller), Loft, 1, 2 eller 3.
- Romnummer er et løpenummer fra 0 og opp
- Pluggnummeret er 1, 2, 3, 4, 5, 6, 7 eller 8.

Eksempel:

- A2-144-3 betyr plugg nummer 3 i rom nummer 144 i 2. etasje av Bygg A.
- CLoft-2-1 betyr plugg nummer 1 i rom nummer 2 på loftet i Bygg C.

Kodene kan ikke være lengre enn 15 tegn.

Vasking er det å fjerne urelevante tegn som bruker kan ha tastet inn. En skal også vaske bort farlige tegn, som kan skape uønskede sikkerhetsproblemer nå eller senere. Den renvaskede koden sendes så til validering.

Validering innebærer å sjekke at koden vil få plass i kolonnen, og at koden har korrekt format, slik beskrevet over. Bruk regulært uttrykk (*regular expression*) for å sjekke om oppgitt kode følger formatreglene.

Om du ikke får til å skrive et regulært uttrykk og validere koden, kan du likevel skrive resten av programmet som legger koden inn i databasen. Husk å sikre mot SQL injection.

Tabellen “opunkt” har bare en kolonne, nemlig id som er en VARCHAR(15) UNIQUE. Programmet skal som en del av valideringen (nevnt over), avvise forsøk på å legge inn koder som er for lange og ikke får plass.

Her er “createdb.sql” som vil opprette databasen, om du ønsker det:

```
DROP DATABASE IF EXISTS byggdata;  
CREATE DATABASE byggdata;  
USE byggdata;  
CREATE TABLE opunkt ( id VARCHAR(15) UNIQUE );
```

Her er hjelpeteksten, som du kan lime inn og formattere i hjelp.php:

Et tilkoblingspunkt er identifisert med en spesiell kode som viser bygning, etasje, romnummer og pluggnummer.

- Bygningskodene er A, B eller C.
- Etasjene er Kj (for kjeller), Loft, 1, 2 eller 3.

- Romnummer er et løpenummer fra 0 og opp
- Pluggnummeret er 1, 2, 3, 4, 5, 6, 7 eller 8.

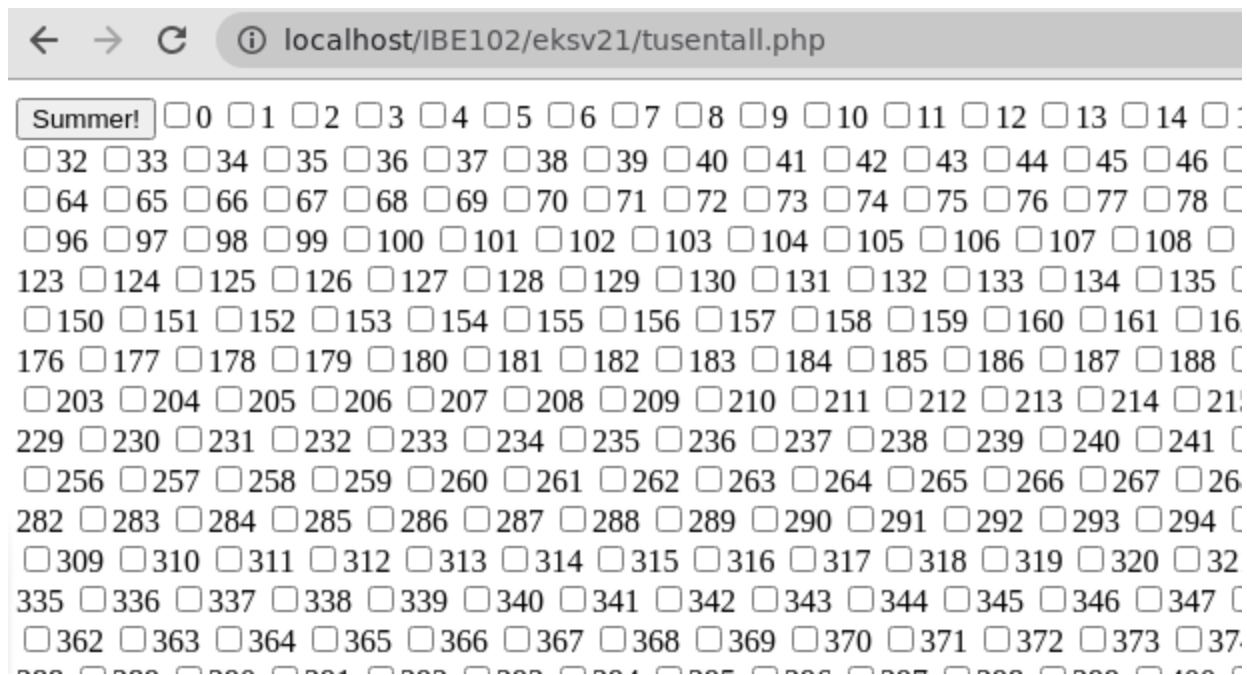
Eksempel:

- A2-144-3 betyr plugg nummer 3 i rom nummer 144 i 2. etasje av Bygg A.
- CLoft-2-1 betyr plugg nummer 1 i rom nummer 2 på loftet i Bygg C.

Kodene kan ikke være lengre enn 15 tegn.

Oppgave 4: Tusen tall (25 %)

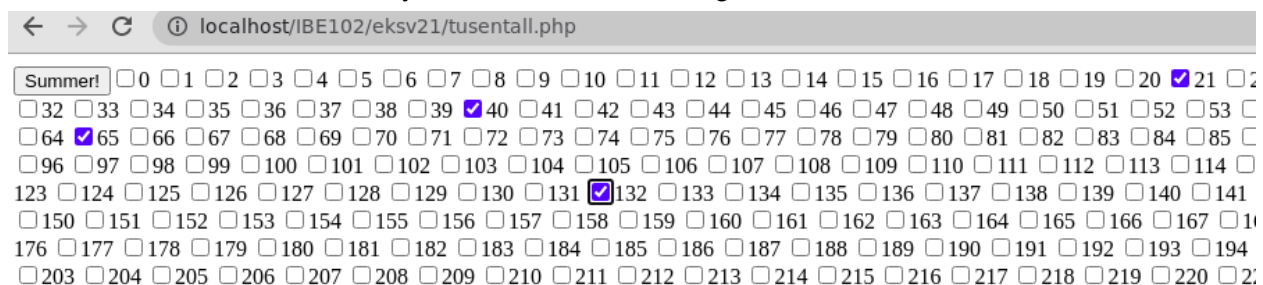
Lag “tusentall.php” som viser tusen tall med hver sin sjekkboks. Her ser vi utsnitt av skjermbildet, når vi starter programmet første gang og ingen er avkrysset:



Sjekkboksene vises fortløpende. De vil fylle skjermbredden og bruke så mange linjer som trengs.

Brukeren kan nå krysse av i en eller flere av disse. Når “Summer!” trykkes vises regnestykket og summen av de han krysset av for.

Under ser vi at brukeren har krysset av for 21, 40, 65 og 132.



Etter man trykker “Summer!” vil skjemaet sendes inn. Neste skjermbilde viser svaret med utregningen øverst, der vi ser alle leddene og summen. Under den horisontale streken vises et blankt skjema med tusen tall og sjekkbokser, og en knapp:

21+40+65+132+=258

Summer! ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21 ☐ 22 ☐ 23 ☐ 24 ☐ 25 ☐ 26 ☐ 27 ☐ 28 ☐ 29 ☐ 30 ☐ 31 ☐ 32 ☐ 33 ☐ 34 ☐ 35 ☐ 36 ☐ 37 ☐ 38 ☐ 39 ☐ 40 ☐ 41 ☐ 42 ☐ 43 ☐ 44 ☐ 45 ☐ 46 ☐ 47 ☐ 48 ☐ 49 ☐ 50 ☐ 51 ☐ 52 ☐ 53 ☐ 54 ☐ 55 ☐ 56 ☐ 57 ☐ 58 ☐ 59 ☐ 60 ☐ 61 ☐ 62 ☐ 63 ☐ 64 ☐ 65 ☐ 66 ☐ 67 ☐ 68 ☐ 69 ☐ 70 ☐ 71 ☐ 72 ☐ 73 ☐ 74 ☐ 75 ☐ 76 ☐ 77 ☐ 78 ☐ 79 ☐ 80 ☐ 81 ☐ 82 ☐ 83 ☐ 84 ☐ 85 ☐ 86 ☐ 87 ☐ 88 ☐ 89 ☐ 90 ☐ 91 ☐ 92 ☐ 93 ☐ 94 ☐ 95 ☐ 96 ☐ 97 ☐ 98 ☐ 99 ☐ 100 ☐ 101 ☐ 102 ☐ 103 ☐ 104 ☐ 105 ☐ 106 ☐ 107 ☐ 108 ☐ 109 ☐ 110 ☐ 111 ☐ 112 ☐ 113 ☐ 114 ☐ 115 ☐ 116 ☐ 117 ☐ 118 ☐ 119 ☐ 120 ☐ 121 ☐ 122 ☐ 123 ☐ 124 ☐ 125 ☐ 126 ☐ 127 ☐ 128 ☐ 129 ☐ 130 ☐ 131 ☐ 132 ☐ 133 ☐ 134 ☐ 135 ☐ 136 ☐ 137 ☐ 138 ☐ 139 ☐ 140 ☐ 141 ☐ 142 ☐ 143 ☐ 144 ☐ 145 ☐ 146 ☐ 147 ☐ 148 ☐ 149 ☐ 150 ☐ 151 ☐ 152 ☐ 153 ☐ 154 ☐ 155 ☐ 156 ☐ 157 ☐ 158 ☐ 159 ☐ 160 ☐ 161 ☐ 162 ☐ 163 ☐ 164 ☐ 165 ☐ 166 ☐ 167 ☐ 168 ☐ 169 ☐ 170 ☐ 171 ☐ 172 ☐ 173 ☐ 174 ☐ 175 ☐ 176 ☐ 177 ☐ 178 ☐ 179 ☐ 180 ☐ 181 ☐ 182 ☐ 183 ☐ 184 ☐ 185 ☐ 186 ☐ 187 ☐ 188 ☐ 189 ☐ 190 ☐ 191 ☐ 192 ☐ 193 ☐ 194 ☐ 195 ☐ 196 ☐ 197 ☐ 198 ☐ 199 ☐ 200 ☐ 201 ☐ 202 ☐ 203 ☐ 204 ☐ 205 ☐ 206 ☐ 207 ☐ 208 ☐ 209 ☐ 210 ☐ 211 ☐ 212 ☐ 213 ☐ 214 ☐ 215 ☐ 216 ☐ 217 ☐ 218 ☐ 219 ☐ 220 ☐ 221 ☐ 222 ☐ 223 ☐ 224 ☐ 225 ☐ 226 ☐ 227 ☐ 228 ☐ 229 ☐ 230 ☐ 231 ☐ 232 ☐ 233 ☐ 234 ☐ 235

For å regne ut summen, må du (selvsagt) vite hvilke tall som ble avkrysset. Et hint til en måte å gjøre dette på, finner du i adresselinjen i eksemplet over med de 4 tallene.

Med sessions som hukommelse: tusentall2

Ta en kopi og kall den “tusentall2.php”, der du legger inn en liten endring: Her skal brukeren ikke kunne velge tall som er valgt tidligere. Hvis han krysser av for eksempel 1 og 7 og sender inn for summering, vil disse i neste skjermbilde ikke lenger vises i listen over tall som kan avkrysses.

Under ser du skjermbildet ved første kjøring, og bruker har valgt seg 1 og 7.

Summer! ☐ 0 ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☒ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21 ☐ 22 ☐ 23 ☐ 24 ☐ 25 ☐ 26 ☐ 27 ☐ 28 ☐ 29 ☐ 30 ☐ 31 ☐ 32 ☐ 33 ☐ 34 ☐ 35 ☐ 36 ☐ 37 ☐ 38 ☐ 39 ☐ 40 ☐ 41 ☐ 42 ☐ 43 ☐ 44 ☐ 45 ☐ 46 ☐ 47 ☐ 48 ☐ 49 ☐ 50 ☐ 51 ☐ 52 ☐ 53 ☐ 54 ☐ 55 ☐ 56 ☐ 57 ☐ 58 ☐ 59 ☐ 60 ☐ 61 ☐ 62 ☐ 63 ☐ 64 ☐ 65 ☐ 66 ☐ 67 ☐ 68 ☐ 69 ☐ 70 ☐ 71 ☐ 72 ☐ 73 ☐ 74 ☐ 75 ☐ 76 ☐ 77 ☐ 78 ☐ 79 ☐ 80 ☐ 81 ☐ 82 ☐ 83 ☐ 84 ☐ 85 ☐ 86 ☐ 87 ☐ 88 ☐ 89 ☐ 90 ☐ 91 ☐ 92 ☐ 93 ☐ 94 ☐ 95 ☐ 96 ☐ 97 ☐ 98 ☐ 99 ☐ 100 ☐ 101 ☐ 102 ☐ 103 ☐ 104 ☐ 105 ☐ 106 ☐ 107 ☐ 108 ☐ 109 ☐ 110 ☐ 111 ☐ 112 ☐ 113 ☐ 114 ☐ 115 ☐ 116 ☐ 117 ☐ 118 ☐ 119 ☐ 120 ☐ 121 ☐ 122 ☐ 123 ☐ 124 ☐ 125 ☐ 126 ☐ 127 ☐ 128 ☐ 129 ☐ 130 ☐ 131 ☐ 132 ☐ 133 ☐ 134 ☐ 135 ☐ 136 ☐ 137 ☐ 138 ☐ 139 ☐ 140 ☐ 141 ☐ 142 ☐ 143 ☐ 144 ☐ 145

Etter at bruker har trykket “Summer!” får vi dette skjermbildet: 1 og 7 er ute av listen:

← → ↻ ⓘ localhost/IBE102/eksv21/tusentall2.php?knapp=Summer%21&1=on&7=on

1+7+=8

Summer! ☐0 ☐2 ☐3 ☐4 ☐5 ☐6 ☐8 ☐9 ☐10 ☐11 ☐12 ☐13 ☐14 ☐15 ☐16 ☐17 ☐18
☐33 ☐34 ☐35 ☐36 ☐37 ☐38 ☐39 ☐40 ☐41 ☐42 ☐43 ☐44 ☐45 ☐46 ☐47 ☐48 ☐49 ☐5
☐65 ☐66 ☐67 ☐68 ☐69 ☐70 ☐71 ☐72 ☐73 ☐74 ☐75 ☐76 ☐77 ☐78 ☐79 ☐80 ☐81 ☐8
☐97 ☐98 ☐99 ☐100 ☐101 ☐102 ☐103 ☐104 ☐105 ☐106 ☐107 ☐108 ☐109 ☐110 ☐111 ☐
☐124 ☐125 ☐126 ☐127 ☐128 ☐129 ☐130 ☐131 ☐132 ☐133 ☐134 ☐135 ☐136 ☐137 ☐13
☐151 ☐152 ☐153 ☐154 ☐155 ☐156 ☐157 ☐158 ☐159 ☐160 ☐161 ☐162 ☐163 ☐164 ☐
☐177 ☐178 ☐179 ☐180 ☐181 ☐182 ☐183 ☐184 ☐185 ☐186 ☐187 ☐188 ☐189 ☐190 ☐19
☐197 ☐198 ☐199 ☐200 ☐201 ☐202 ☐203 ☐204 ☐205 ☐206 ☐207 ☐208 ☐209 ☐210

Selve utregningen som vises over streken, er uforandret. Den er summen av de tallene som ble valgt i *siste runde*. Det er altså ikke summen av alle tidligere valgte tall.

Så, om vi ser brukeren velge seg to nye tall:

← → ↻ ⓘ localhost/IBE102/eksv21/tusentall2.php?knapp=Summer%21&1=on&7=on

1+7+=8

Summer! ☐0 ☐1 ☐2 ☐3 ☐4 ☐5
☐25 ☐26 ☐27 ☐28 ☐29 ☐30
☐49 ☐50 ☐51 ☐52 ☐53 ☐54
☐73 ☐74 ☐75 ☐76 ☐77 ☐78
☐97 ☐98 ☐99 ☒100 ☐101 ☐102
☐118 ☐119 ☒120 ☐121 ☐122
☐138 ☐139 ☐140 ☐141 ☐142
☐159 ☐160 ☐161 ☐162 ☐163

Så er altså utregningen gjort på kun de siste valgte tallene. I neste skjermbilde ser vi derfor dette: 100 og 120 address, og vi ser at 100 og 120 er ute av listen over valgbare tall (sammen med 1 og 7 fra tidligere).

← → ↻ ⓘ localhost/IBE102/eksv21/tusentall2.p

100+120+=220

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐8 ☐9 ☐10
☐25 ☐26 ☐27 ☐28 ☐29 ☐30 ☐31 ☐32 ☐33 ☐34
☐49 ☐50 ☐51 ☐52 ☐53 ☐54 ☐55 ☐56 ☐57 ☐58
☐73 ☐74 ☐75 ☐76 ☐77 ☐78 ☐79 ☐80 ☐81 ☐82
☐97 ☐98 ☐99 ☐101 ☐102 ☐103 ☐104 ☐105 ☐10
☐119 ☐121 ☐122 ☐123 ☐124 ☐125 ☐126 ☐127 ☐
☐140 ☐141 ☐142 ☐143 ☐144 ☐145 ☐146 ☐147 ☐
☐160 ☐161 ☐162 ☐163 ☐164 ☐165 ☐166 ☐167 ☐
☐180 ☐181 ☐182 ☐183 ☐184 ☐185 ☐186 ☐187 ☐

Bruk sessions for å realisere den nødvendige hukommelse. En mulighet (dvs et hint): Husk tidligere tall i en string "valgtetall", som du utvider etter hvert, og så søker i med funksjonen strpos. I eksemplet over vil stringen etter første runde inneholde "1 7", og etter andre runde inneholde "1 7 100 120".

Lever dette som "tusentall2.php".

(slutt eksamenstekst).