

Tenta i Grundläggande programmering
DD1331 2017-10-20 klockan 14.00–16.00

Tillåtna hjälpmedel: En Pythonbok, skrivmaterial, mat, medicin och vattenflaska.

Otillåtna hjälpmedel: Laptop, mobiltelefon eller miniräknare.

Bonuspoäng (upp till 4 för labbar redovisade i tid) gäller. För godkänt krävs 15 poäng. Högsta poängen du kan få är 24 (20 tentapoäng och 4 bonuspoäng).

Hur du fyller i dina svar

Skriv dina svar på ett separat svarspapper. Skriv namn och personnummer överst på det papperet. För varje uppgift, skriv en rad med uppgiftens nummer till vänster följt av det eller de alternativ som du tror är rätt svar. Skriv med stora bokstäver. När tentan är klar lämnar du bara in ditt svarspapper.

Exempel:

1. P, Å
2. Y
3. T, Y
4. H, O, P
5. O, S, T
6. N, P

Tenta i DD1331 Grundläggande Programmering DD1331

1. (1p)

Följande Python3-uttryck evalueras i Idle. Vad blir resultatet?

```
>>> "PARIS"[:2]
```

- (A) 'A'
- (B) 'AI'
- (C) 'PA'
- (D) 'PAR'
- (E) 'PRS'
- (F) 'R'
- (G) 'RIS'
- (H) 'SIRAP'
- (I) 'SPIRA'

2. (1p)

Följande Python3-uttryck evalueras i Idle. Vad blir resultatet?

```
>>> 1/2/2
```

- (A) 0
- (B) 0.0
- (C) 0.25
- (D) 1
- (E) 1.0
- (F) Inget av ovanstående alternativ.

3. (2p)

Vad skriver detta program ut?

```
def mystery(dygder):  
    dygder[1] = dygder[2]  
  
def main():  
    dygder = ["rättvisa", "tapperhet", "klokhets", "måttfullhet"]  
    mystery(dygder)  
    for x in dygder:  
        print(x, end = " ")  
    print()  
  
main()
```

- (A) tapperhet rättvisa klokhets måttfullhet
- (B) tapperhet tapperhet klokhets måttfullhet
- (C) rättvisa klokhets tapperhet måttfullhet
- (D) rättvisa klokhets klokhets måttfullhet
- (E) klokhets måttfullhet rättvisa tapperhet

4. (1p)

Vad svarar Idle efter att dessa rader Python3 har matats in?

```
>>> lista = [3,1,4,1,5,9,2,6]
>>> lista[2]+lista[-2]
```

- (A) 1
- (B) 2
- (C) 3
- (D) 4
- (E) 5
- (F) 6
- (G) 7
- (H) 8
- (I) 9
- (J) Inget av ovanstående alternativ.

5. (2p)

Vilket/Vilka alternativ beskriver bäst vad detta Python3-program gör?

```
def mystery():
    mystery()
    print("Rekursion")
```

```
mystery()
```

- (A) Det skriver "Rekursion" exakt en gång.
- (B) Raden med print körs inte en enda gång.
- (C) Det skriver "Rekursion" om och om igen tills någon slår av det.
- (D) Det kraschar.
- (E) Det skriver "Rekursion" färre än 100 gånger och avslutas sedan utan att krascha.
- (F) Inget av ovanstående alternativ.

6. (2p)

Vad skriver följande Python3-program ut?

```
def mystery(a,b):  
    a,b = b,a  
    print(a,b, end = " ")
```

```
def main():  
    a,b = 42,13  
    mystery(a,b)  
    print(a,b)
```

```
main()
```

- (A) 13 42 13 42
- (B) 13 13 13 13
- (C) 42 42 42 42
- (D) 13 42 42 13
- (E) 42 13 13 42
- (F) 42 13 42 13

7. (2p)

Vad skriver följande Python3-program ut?

```
def mystery(numbers):  
    numbers[1] = numbers[2]  
    print(numbers, end=" ")
```

```
def main():  
    numbers = [1,2,3,4]  
    mystery(numbers)  
    print(numbers)
```

```
main()
```

- (A) [1, 3, 2, 4] [1, 2, 3, 4]
- (B) [1, 3, 2, 4] [1, 3, 2, 4]
- (C) [1, 3, 3, 4] [1, 2, 3, 4]
- (D) [1, 3, 3, 4] [1, 3, 3, 4]
- (E) [2, 1, 3, 4] [1, 2, 3, 4]
- (F) [2, 1, 3, 4] [2, 1, 3, 4]
- (G) [2, 2, 3, 4] [1, 2, 3, 4]
- (H) [2, 2, 3, 4] [2, 2, 3, 4]

8. (2p)

Byt ut kommentaren i koden nedan mot ett svarsalternativ i taget. Vilket/vilka alternativ får detta Python3-program att exekvera raden med print?

```
def main():  
    # Här ska svaret stoppas in i koden.  
    if ( not a or b) and (not b or c):  
        print(a,b,c)
```

main()

- (A) a, b, c = True, True, True
- (B) a, b, c = True, True, False
- (C) a, b, c = True, False, True
- (D) a, b, c = True, False, False
- (E) a, b, c = False, True, True
- (F) a, b, c = False, True, False
- (G) a, b, c = False, False, True
- (H) a, b, c = False, False, False
- (I) Inget av ovanstående alternativ.

9. (1p)

En sorterad lista innehåller 250 heltal. Du vill undersöka om talet 997 finns med i listan och använder binärsökning. Hur många tal i listan kommer binärsökningen maximalt att läsa av?

- (A) 2 element
- (B) Ungefär 4 element
- (C) Ungefär 8 element
- (D) Ungefär 16 element
- (E) Ungefär 32 element
- (F) Ungefär 64 element
- (G) Ungefär 128 element
- (H) Alla 256 elementen

10. (1p)

Vad är det sista som programmet skriver ut innan det avslutas?

```
def main():  
    for i in range(5):  
        print(i)
```

```
main()
```

- (A) 1
- (B) 2
- (C) 3
- (D) 4
- (E) 5
- (F) 6
- (G) 7
- (H) Inget av ovanstående alternativ.

11. (1p)

Vad skriver följande Python3-program ut?

```
def mystery(n):  
    return n*mystery(n-2)
```

```
print(mystery(3))
```

- (A) 0
- (B) 1
- (C) 3
- (D) 6
- (E) Ingenting alls. Det har ett syntaxfel.
- (F) Ingenting alls. Programmet fortsätter för evigt.
- (G) Programmet kraschar.

12. (1p)

Vad skriver följande Python3-program ut?

```
def mystery(name):
    name += " " + "Yousafzai"
    print(name, end = " ")

def main():
    name = "Malala"
    mystery(name)
    print(name)

main()
```

- (A) Malala Yousafzai Malala
- (B) Malala Yousafzai Malala Yousafzai
- (C) Yousafzai Malala
- (D) Malala Yousafzai
- (E) Malala Yousafzai Yousafzai

13. (1p)

Följande kod implementerar insertion sort (insättningssortering). Vad är tidskomplexiteten i värsta fallet (mätt i antalet element i listan)?

```
def insertion_sort(v):
    """Changes v to a sorted list."""
    for index, value in enumerate(v):
        if index == 0 or v[index-1] <= value:
            continue
        for push_index in range(index-1, -1, -1):
            if v[push_index] <= value:
                push_index += 1
                break
        v[push_index+1] = v[push_index]
        v[push_index] = value
    return v
```

- (A) logaritmisk: $O(\log(n))$
- (B) linjär: $O(n)$
- (C) $n \cdot \log(n)$: $O(n * \log(n))$
- (D) kvadratisk: $O(n^2)$

14. (1p)

Vilket Bash-kommando är bäst för att skriva ut de rader i en textfil som innehåller strängen "mod", men inte rader som inte innehåller den strängen?

- (A) assert
- (B) cp
- (C) grep
- (D) mv
- (E) raise
- (F) rm
- (G) wc

15. (1p)

Vilket/Vilka av följande påståenden om Github stämmer?

- (A) Du behöver lägga upp din privata SSH-nyckel på Github innan du kan pusha ändringar till en utvecklingskatalog/ett githubrepo.
- (B) git commit används för att tala om för git vilka filer som ska ändras.
- (C) git add används för att lägga in en kommentar om vilka ändringar som genomförts.
- (D) Det enda du behöver göra innan git push är att spara filen som har ändrats.
- (E) Inget av ovanstående.