

Lösningar och kommentarer för tentamen 2026-03-14 i DA2005

Del A: flervalsfrågor

1. E
2. B, C, D, E
3. C
4. B
5. C
6. A, B, D, E
7. D
8. C

Del B: kodfrågor

9. Exempellösningar:

A.

```
def add_positive(ns):
    s = 0

    for n in ns:
        if n > 0:
            s += n

    return s
```

B.

```
l = []
while True:
    try:
        n = int(input("Write a number: "))
        if n == 0:
            print("Sum of positive numbers: " + str(add_positive(l)))
            break
        elif n > 0:
            l.append(n)
        else:
            pass
    except ValueError:
        print("That is not a number!")
```

10. Exempellösningar:

A.

```
def to_dict(shopping_list):
    out = {}

    for s in shopping_list:
        if s in out:
            out[s] += 1
        else:
            out[s] = 1

    return out
```

B.

```
def dict_from_file(f):
    with open(f) as in_file:
        d = {}
        for r in in_file:
```

```

        x,n = r.split()
        if x in d:
            d[x] += int(n)
        else:
            d[x] = int(n)
    return d

```

11. Objektorienterad exempellösning:

```

class Item():

    def __init__(self,name,y,m,d):
        self.name = name
        self.best_before = (y,m,d)

    def __str__(self):
        y,m,d = self.best_before
        return self.name + " (best before: " + str(d) + "/" + str(m) + " " + str(y) + ")"

    def is_still_ok(self,y,m,d):
        yb,mb,db = self.best_before

        if y > yb:
            return False
        elif y == yb and m > mb:
            return False
        elif y == yb and m == mb and d > db:
            return False
        else:
            return True

```

Del C: kodfrågor inspirerade av labbarna

12. Exempellösningar:

A.

```

def celsius_to_rankine(t):
    return 1.8 * (t + 273)

def rankine_to_celsius(t):
    return t / 1.8 - 273

```

B.

```

ans = input("Which conversion do you want to do? ")

if ans == "Rankine to Celsius":
    t = float(input("Write temperature in Rankine to convert: "))
    print("Temperature in Celsius: ",str(rankine_to_celsius(t)))
elif ans == "Celsius to Rankine":
    t = float(input("Write temperatue in Celsius to convert: "))
    print("Temperature in Rankine: ",str(celsius_to_rankine(t)))
else:
    print("You must specify which conversion you want to do")

```

13. Exempellösningar:

A. För att sortera listan [2,1,4,3] kommer algoritmen först plocka ut min-värdet, vilket är 1, och flytta det sist i en ny lista. Listorna är nu [2,4,3] och [1]. Sedan kommer detta repeteras så att listorna är [4,3] och [1,2], sedan [4] och [1,2,3], och slutligen [] och [1,2,3,4]. Den sorterade listan kommer då returneras eftersom den alla element sorterats (dvs den osorterade indatalistan är tom).

B.

```
def selection_sort(l):
    out = []

    while (l != []):
        m = min(l)
        out.append(m)
        l.remove(m)

    return out
```

14. Exempellösningar:

A. Algoritmen kommer först börja med att sätta räknaren till 0 då inga element sorterats. Sedan kommer den hitta index på min-värdet i den osorterade listan (dvs hela listan i första iterationen). Detta är 1 på index 1. Algoritmen byter då plats på värdena på index 0 och 1, vilket ger [1,2,4,3] och räknaren uppdateras till 1. Algoritmen fortsätter sedan, eftersom 2 är på rätt plats kommer inget hända. I nästa iteration är listan [1,2,4,3] och index 2. Algoritmen hittar då nya min-värdet vilken är 3 på index 3, detta byts då ut mot värdet på index 2 och listan är [1,2,3,4]. Vi är sedan på sista elementet och inget kommer då bytas ut då vi inte har några element att jämföra med. Sedan är index samma som längden på listan och algoritmen är då klar och listan är sorterad.

B.

```
def selection_sort_in_place(l):
    i = 0
    ln = len(l)

    while (i < ln):

        # find new min and its position, starting from index i
        new_min = l[i]
        pos_new_min = i

        # find the true new min and its position
        for j in range(i, ln):
            if l[j] < new_min:
                new_min = l[j]
                pos_new_min = j

        # swap the values at index i and pos_new_min
        l[pos_new_min] = l[i]
        l[i] = new_min

        # continue looping
        i += 1

    return l
```