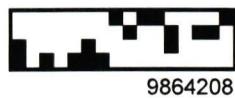




2-U6A5M84V6FKQGP4



9864208

FÖRSÄTTSBLAD TENTAMEN/ EXAMINATION COVER

Jag intygar att mobiltelefon och annan otillåten elektronisk utrustning är avstängd och förvaras på anvisad plats. / I hereby confirm that mobile phones and other unauthorized electronic equipment is shut off and placed according to instructions

MARKERA MED "X" /
MARK WITH "X"

IFYLLES AV STUDENT OCH TENTAMENSVAKT/
TO BE FILLED IN BY THE STUDENT AND THE INVIGILATOR:

KURSKOD / COURSE CODE <table border="1"><tr><td>I</td><td>D</td><td>1</td><td>0</td><td>1</td><td>8</td></tr></table>	I	D	1	0	1	8	EFTERNAMN / FAMILY NAME KC
I	D	1	0	1	8		
KURSNAMN / COURSE NAME Programmering I	FÖRNAMEN / FIRST NAME Sushil						
PROVKOD / TEST CODE <table border="1"><tr><td>T</td><td>E</td><td>N</td><td>1</td></tr></table>	T	E	N	1	NAMNTECKNING / YOUR SIGNATURE Sushil ke		
T	E	N	1				
TENTAMENSDATUM / EXAMINATION DATE Y/Y/Y/Y M/M D/D 2022 - 01 - 14		PERSONNUMMER / PERSONAL NUMBER Y/Y/Y/Y/M/M/D/D 20020418 - 5052					
PROGRAMKOD / PROGRAM CODE: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	INLÄMNINGSTID / TIME SUBMITTED: 17 ⁴⁸	SIGNUR TENTAMENSVAKT / SIGNATURE INVIGILATOR: 	ANTAL BLAD / NO OF SHEETS: 08				
MARKERA BEHANDLADE UPPGIFTER MED "X" OCH EJ BEHANDLADE UPPGIFTER MED "-" / MARK WITH "X" PROBLEMS SOLVED. MARK WITH "-" PROBLEMS NOT ATTEMPTED							
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	X X X X X						

IFYLLES AV INSTITUTIONEN / TO BE FILLED IN BY THE DEPARTMENT:

BEDÖMNING / ASSESSMENT																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20							
BONUSPOÄNG/ BONUS POINTS: <table border="1"><tr><td></td><td></td></tr></table> , <table border="1"><tr><td></td></tr></table>								SLUTSUMMA / FINAL POINTS: <table border="1"><tr><td></td><td></td></tr></table> , <table border="1"><tr><td></td></tr></table>								BETYG/ GRADE: <table border="1"><tr><td></td></tr></table>										

9864208



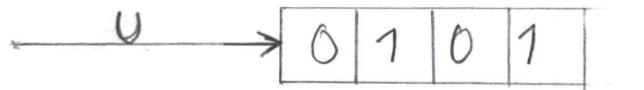
Godkänts av examinator /
approved by Examiner.....



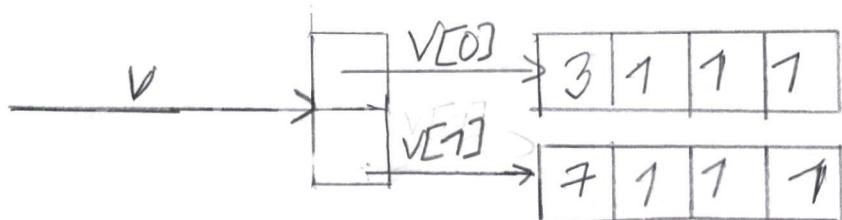
Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
Sushil	0204185052		1	1

Uppgift 1

a)



b)





Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
SUSHIL	0204185052		2	2

Uppgift 2

a)

```
public static Point+ nearestPoint (Point+[] points)
{
    double nearest = points[0].distance (ORIGIN);
    int nearestPointIndex = 0;
    for (int i=1; i < points.length; i++)
    {
        if (nearest > points[i].distance (ORIGIN))
        {
            nearest = points[i].distance (ORIGIN);
            nearestPointIndex = i;
        }
    }
    return points[nearestPointIndex];
}
```

b)

```
public static Point+[] selectPoints (Point+[] points, Point+ valdpunkt)
{
    int antalpunkter = 0;
    for (int i=0; i < points.length; i++)
    {
        if (valdpunkt.distance (points[i]) < 4)
        {
            antalpunkter++;
        }
    }
    Point+[] chosenPoints = new Point[antalpunkter];
    int n = 0;
    int pos = 0;
    while (n < antalpunkter)
    {
        if (valdpunkt.distance (points[pos]) < 4)
        {
            chosenPoints[n] = points[pos];
            pos++;
        }
        n++;
    }
}
```



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
Sushil	200204185052		3	2.

b) ArrayList chosen

$\text{return chosenpoints;}$

c) $\text{Point[] points = \{new Point(2.0, 3.0),}$
 $\text{new Point(3.5, 3.0),}$
 $\text{new Point(8.0, 8.5),}$
 $\text{new Point(9.0, 2.0),}$
 $\text{new Point(1.0, 1.0),}$
 $\text{new Point(4.0, 5.0)\};}$

$\text{Point aPoint = new Point(6.0, 8.0);}$

$\text{Point theNearestPoint = nearestPoint(points);}$

$\text{Point[] valdappnletor = selectPoints(points, aPoint);}$



Family name, first name
Sushil

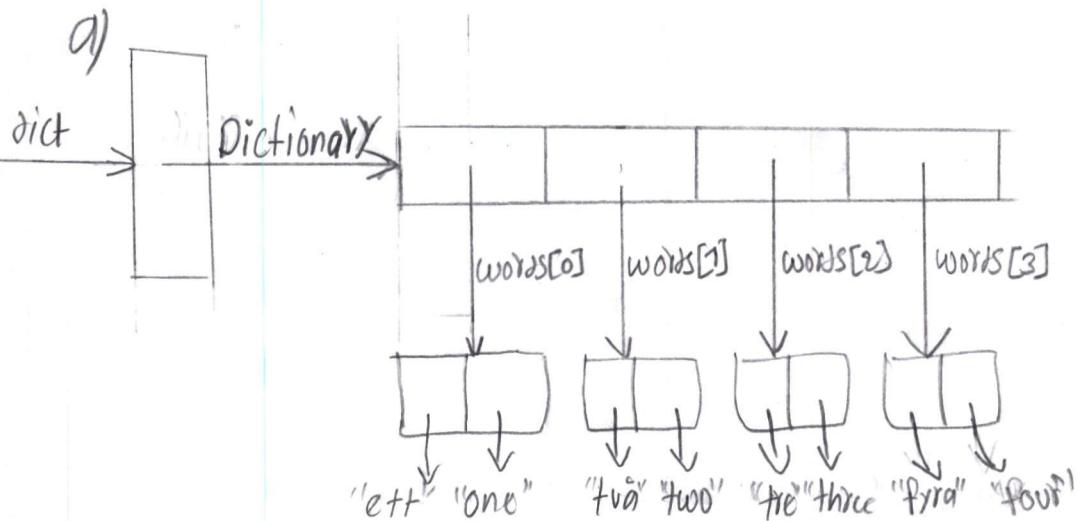
Personal Registration Number
0204185052

Programme

Sheet no.
4

Problem no.
3

Uppgift 3



b)

```
public void addWord(String sWord, String eWord)
{
    int i=0;
    Word[] addWords = new Word[words.length+1];
    for( i=0; i<words.length; i++)
    {
        addWords[i]= words[i];
    }
    addWords[i]= new Word(sWord, eWord);
    words= addWords;
}
```

① ~~public String translate(String sWord) throws IllegalArgument~~
② -Exception

③ int index = -1;
④ if (index < words.length; i++)
⑤ for (int i=0; i<words.length; i++)
⑥ { if (words[i].sWord.equals(sWord))
⑦ { index = i;
⑧ break; } }
⑨ if (index == -1)
⑩ throw new IllegalArgumentException("Exception(" + word + " not found in words);")
⑪ return word; } catch (Exception e) {



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
Sushil	0209185052		5	3

c)

public String translate (String sWord) {

throws IllegalArgumentException

}

int index = -1;

for (int i = 0; i < words.length; i++)

{ if (words[i].sWord.equals(sWord))

{ index = i;
break;

}

if (index == -1)

throw new java.lang.IllegalArgumentException("word not found:
+sWord);

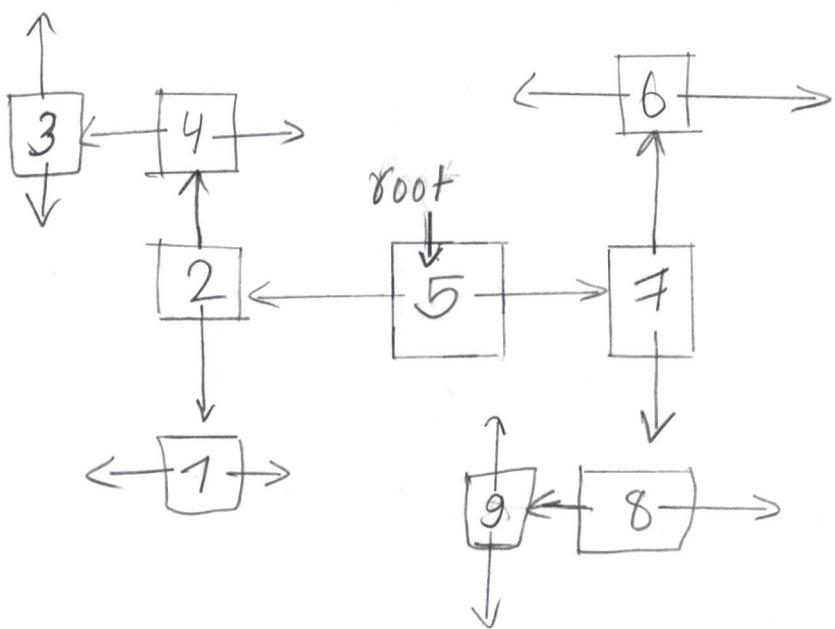
return words[index].eWord;

{

Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
SUSHIL	0204185052		6	4

a)

Uppgift 4



b)

2 4 3

5 ≠ 18 9



Family name, first name

SUSHIL

UPPGIFT 5

Personal Registration Number

0204185052

Programme

Sheet no.

7

Problem no.

5

a) 54213

45213

24513

12453

12345

b)

Antalet elementjämförelser när det gäller värska fall är när element sitter i en omvänt ordning. När det gäller värska fall jämför denna algoritm det första/tråd element en gång sen den tredje element 2 gånger osv. För varje n^{th} element efter n^{th} okas det också antalet jämförelser men ett för att bestämma var det sista element ska ligga jämför det $n-1$ gånger för näst sista, $n-2$ osv. Antalet element jämförelser kan skrivas så här:

$$1 + 2 + \dots + (n-2) + (n-1)$$

Detta ger komplexitetsfunktionen:

$$W(n) = n \frac{(n-1)}{2}$$

Detta kan också skrivas så:

$$W(n) = \frac{n^2}{2} - \frac{n}{2}$$

för storan n domineras termen n^2 och funktionen tillhör $W(n) \in \Theta(n^2)$

Algoritmen är kvadratisk när det gäller element jämförelser i värska fall.

~~När det gäller bästa fall är element redan sorterade och algoritmen jämför de första två elementen en gång och för varje ny element jämföras en gång. Den totala jämförelserna skrivs så:~~

$$B(n) = (n-1)$$

~~för tillräckligt stora antal element domineras termen n och komplexitetsfunktionen tillhör:~~

$$B(n) \in \Theta(n)$$

~~Algoritmen är linjär när det gäller antalet jämförelser i bästa fall.~~



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
Sushil	200209185052		8	5

när det gäller bästa fall är element redan sorterade och algoritmen jämför de första två talen en gång om för varje ny tal kommer det att sättas i jämförelser att öka men ett om den totala jämförelser kan skrivas så:

$$B(n) = (n-1)$$

för tillräcklig stora anta element domineras termen n och komplexitetsfunktion tillhör:

$$B(n) \in \Theta(n)$$

Algoritmen är linjär när det gäller antalet jämförelser i bästa fall.