



Antal blad /
Number of sheets

12 ✓

TENTAMEN / EXAMINATION

Anvisningar:

Skriv din anonymitetskod på varje blad.
Endast en uppgift får lösas på varje blad.
Var vänlig skriv tydligt!

Instructions:

Write your anonymous code on each sheet.
Answer only one question on each sheet.
Please write clearly!

Vänligen texta anonymitetskoden i textboxen enligt exempel nedan!
Please write the Anonymous Code clearly in the textbox like example below!

Bokstäver/Letters:

A-B-C-D-E-F-G-H-I-J-K-L-M-N-O
P-Q-R-S-T-U-V-W-X-Y-Z-Å-Ä-Ö

Siffror/Numbers:

Ø-1-2-3-4-5-6-7-8-9

Exempel:

A	B	C	1	7	Ø	-	Ø	1	7
---	---	---	---	---	---	---	---	---	---

DVGA01 Programmeringsteknik

Kurskod + Kurs / Course Code + Course:

Delkurs / Part course:

Anonymitetskod / Anonymous code = Kurskod + kodnr / course code + code number									
D	V	G	A	Ø	1	-	Ø	1	3

Tentamensdatum / Examination date:
31/10

Behandlade uppgifter / Solved problems

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X	X	X	X	X	X	X	X	X						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Ifylles av lärare / To be completed by the examiner

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	10	5	4	5	5	4	4	5						
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Poäng / Marks gained: 47

Betyg / Grade: 5

Max poäng / Total marks gained: 50

För Gk poäng / Marks gained to be passed: 25

Marki Blom
Examin. lärare / Kursansvarig signatur / Signature of the examiner

Namnförtydligande / Clarification of the signature



1

a, En main måste varje program ha. Main är en funktion som körs först av alla funktioner

ex. `int dubbla()`

`int main()`

Här körs main först. |

b, En algoritm är en tydlig beskrivning på hur man löser ett problem.

Ett exempel är ett recept som tydligen visar vad du måste göra för att t.ex. baka en kaka. |

c, Selektion, används så att programmet går en viss väg om vissa saker stämmer

t.ex. `if(x > 1)`
`printf("x är större än ett");`

Här skrivs x är större än ett ut om x är större än ett |



1, d, Vi har en funktion
`int dubbla(int x)`

Där är `int x` den formella
variabeln. Alltså hamnar
den lokala variabeln för.

e, Det är en variabel som
skapas och dörs i en
funktion.

t.ex.

```
int dubbla(int x)
{
    int dubblad;
    dubblad = (2 * x);
    return 0;
}
```

Här är `dubblad` en lokal
variabel som försviner
när funktionen är slut.



2

a, När syftet inte är att returnera någonting är det bra.

Finns massor olika funktioner för en funktion med void som returtyp, ett exempel är när man vill skriva ut en meny. Ett annat kan ju vara att man använder sig av pekare så man inte behöver returnera något

2

b, När du gör ett värdeanrop tar du värdet av en variabel, kopierar det och sedan gör vad du vill med det. Vid referensanrop ändras du värdet på den variabel du har som aktuell parameter.

2



2, c, Syftet med .h filer, alltså headerfiler är att sammankoppla olika .c filer. Samt att ha ett ställe att ha alla includes och defines på. 2

d, tiring nesting is 2

e, Den har inget returne. 2

Min gcc varnar inte ens.

minneslekage



Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DUGA01 - 013

Löpande sidnr
Consecutive no:

5

Uppgift nr /
Question no:

3

Poäng / Points
awarded:

5

Lärarens
anteckning
Examiner's remarks:

3, a

```
int main()
```

```
{ char a = 'A';
```

```
char b = 'Z';
```

```
for(a; a <= b; a++)
```

```
{
```

```
printf("%c\t%d\n", a, a);
```

```
}
```

```
}
```

b, A=5, B=3, C=15, D=7

4,

A: K2=5

B: K2=5

C: K2=5

D: K2=5

E: K2=5

3

2

Häftområde

Skriv ej i detta område
Leave this area blank



Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DVGA01-073

Löpande sidnr
Consecutive no:

6

Uppgift nr /
Question no:

4

Poäng / Points
awarded:

4

Lärarens
anteckning
Examiner's remarks:

4r

$$funkt_2 = 5$$

$$funkt_3 = 5$$

$$funkt_2 = 7$$

$$funkt_1 = 19$$

Resultat 44 close, but no cigar :)
36

Häftområde

Skriv ej i detta område
Leave this area blank



5,

```
summa(int arr[], int size)
{
    int i, summa;
    for(i=0; i<size; i++)
    {
        summa += arr[i];
    }
    return summa;
}
```

```
int main
{
    int size=5;
    int vek[size] = {1,2,3,4,5}
    int summan;
    summan = summa(vek, size);
}
```




Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DUGA01-013

Löpande sidnr
Consecutive no:

8

Uppgift nr /
Question no:

6

Poäng / Points
awarded:

Lärarens
anteckning
Examiner's remarks:

```

6, int main()
{
    int size=25, size2, i;
    char fnamn[size]
    char enamn[size]
    printf("Skriv in ditt fornamn");
    scanf("%s", fnamn);
    printf("Skriv in ditt efternamn");
    scanf("%s", enamn);
    strcat(fnamn, enamn);
    size2 = strlen(fnamn);
    for(i=0; i<size2; i++)
    {
        vocal(fnamn[i]);
    }
    return 0;
}

```



Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DVGAØ1-Ø13

Löpande sidnr
Consecutive no:

9

Uppgift nr /
Question no:

6

Poäng / Points
awarded:

7

Lärarens
anteckning
Examiner's remarks:

```

6
void vowel(char x)
{
    int i, found;
    char vowel[5] = {'a', 'o', 'e', 'i', 'y'};
    for(i=0; i<5; i++)
    {
        if(x == vowel[i])
            found = 1;
        else
            found = 0;
    }
    if(found == 1)
        printf("%c", x);
    else
        printf("%c o %c", x, x);
}

```



Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DVG A01-013

Löpande sidnr
Consecutive no:

10

Uppgift nr /
Question no:

7

Poäng / Points
awarded:

4

Läarens
anteckning
Examiner's remarks:

```
int iteration(int a, int startPos, int antal)
7, int i, summe;
for (i = startPos; i <= antal; i++)
    summe += a[i];
```




```
#Define LENGTH 20

8 a, typedef struct Bok {
    char Titel[LENGTH];
    char forfattare[LENGTH];
    int year;
    int sidor;
} Bok;

b, void spara (Bok x)
    FILE *fp;
    fp = fopen("register.txt", "ab");
    if (fp)
        fwrite(&x, sizeof(Bok), 1,
        fp);
    else
        printf("filen gick inte att öppna");
    fclose(fp);

c, void read()
    FILE *fp;
    Bok x;
    fp = fopen("register.txt", "rb");
    if (fp)
        fread(&x, sizeof(Bok), 1, fp);
        printf("%s, %s, %d, %d\n",
        x.Titel, x.forfattare, x.year, x.sidor);
    else
        printf("filen kunde inte öppnas");
```



```

a, a, void multi-varde (char vek[])
    int size, i;
    char key;
    scanf("%c", &key);
    size = strlen(vek);
    for (i=0; i < size; i++)
    {
        if (key == vek[i])
        {
            printf("Tecknet finns på  
plats %.d\n", i) }
    }
  
```

2

```

b, void finns-storre (int vek[], int
    key, int size)
{
    int i, storsta=0, temp;
  
```

Uag antar att vi vet hur lång
arrayen är.

```

    for (i=0; i < size; i++)
    {
        temp = vek[i]
        if (temp > storsta)
            storsta = temp;
    }
  
```

```

    if (storsta > key)
        printf("Ja, det gör det");
    else
        printf("Nej, det gör det inte");
  
```

3