

1) (1p) Skriv formeln nedan som en tilldelningsats i C.

$$x = \frac{y}{z} + 2\sqrt{2 + y^2}$$

2) (1p) Vilka värden har följande uttryck?

a) $12 / 5 \% 8;$

b) $9 \% 5 / 2;$

3) (1p) Datatypen

```
struct bil
{
    char regnr[10];           // Registreringsnummer
    float effekt;            // Motorns effekt
    int nr_cyl;              // Antal cylindrar
};
```

avbildar bilar. Definiera en variabel av denna typ och initiera den med registreringsnummer ESA112, effekt 145 och antal cylindrar 4.

4) (1p) Ange värdet på x då följande kod körts.

```
x = 12;
while(x >= 0)
{
    if (x == 12 || x == 6 || x == 3)
        x -= 2;
    else
        x--;
}
```

5) (1p) Strängar skickas ofta vid kommunikation mellan dator och mätinstrument. Hur gör man om mätsträngen "5.1" till det reella talet 5.1? Visa omvandlingen med en allmän algoritm som gäller för alla reella tal med en decimal mellan 0.0 och 9.9. Inga färdiga funktioner får användas.

6) (2p) Skriv en funktion, trippel, som tar ett sex-siffrigt positivt heltal som parameter och kontrollerar om talet består av tre likadana tvåsiffriga talkombinationer som ex 252525. Har den det ska funktionen returnera sant (1) annars falskt (0). Funktionshuvud enligt:

```
int trippel(int ptal);
```

- 7) (2p) Skriv en funktion, `visa_bil`, som tar en bil enligt uppgift 3 ovan som parameter och med ledtexter visar dess data på skärmen. Funktionshuvud enligt:

```
void visa_bil(struct bil b);
```

- 8) (2p) Skriv en funktion som kontrollerar att de tre första tecknen i ett registreringsnummer är stora bokstäver A – Z och de tre resterande siffror 0 – 9. Du behöver ej bry dig om Å, Ä eller Ö. Funktionshuvud enligt:

```
int is_ok_regnr(char *regnr);
```

- 9) (2p) Skriv en funktion, `besiktning` som tar ett registreringsnummer enligt uppgift 8 ovan som parameter och returnerar besiktningsmånaden som vi antar är lika med den sista siffran i registreringsnumret + 1 om den sista siffran ligger mellan 0 och 5. Ligger den sista siffran mellan 6 och 9 är besiktningsmånaden sista siffran + 2. Funktionshuvud enligt:

```
int besiktning(char *regnr);
```

- 10) (2p) Hur omvandlar man en sträng till ett reellt tal generellt, alltså som i uppgift 5 ovan men man har en sträng som består av ett antal siffror, en decimalpunkt och ett antal siffror. Skriv en funktion, `str_to_float` som tar en sträng som parameter och som returnerar det reella tal som bildas av strängen. Inga färdiga funktioner får användas och funktionshuvud enligt:

```
float str_to_float(char *str);
```

- 11) (5p) Skriv ett program som läser in två positiva hela tal och som skriver ut alla tal från och med det minsta inlästa talet till och med det största inlästa talet och skriver ut summan av alla utskrivna tal. Två körexempel där du matar in det understrukna:

```
Ge första talet : 21
Ge sista : 25
21 + 22 + 23 + 24 + 25 = 115
```

```
Ge första talet : 11
Ge sista talet : 5
5 + 6 + 7 + 8 + 9 + 10 + 11 = 56
```

- 12) (5p) Skriv ett program som börjar med att fråga efter antalet (max 10) reella mätvärden som ska läsas in, läser in mätvärdena till en reell vektor och skriver ut de mätvärden som avviker mer än 5 % från mätvärdenas medelvärde. Programmet ska upprepa och avslutas direkt då antalet mätvärden anges till 0.

- 13) (5p) Skriv ett program som läser in ett registreringsnummer i form av en sträng och genom att anropa funktionen i uppgift 8 ovan kontrollerar att de tre första tecknen är stora

bokstäver och resten siffror. Programmet ska vara ett filter som inte ger sig utan frågar efter ett nytt registreringsnummer så länge det inlästa är felaktigt. När programmet läst in ett korrekt registreringsnummer ska den skriva ut i vilken månad den ska besiktas genom att anropa funktionen `besiktning` i uppgift 9 ovan. Månaden ska skrivas ut i klartext som januari eller februari etc.

14(5p) Textfilen `Bilar.txt` innehåller ett antal bilar av typen `struct bil` enligt uppgift 3 ovan med registreringsnummer, effekt och antal cylindrar radvis enligt :

```
acd123      132    6
wer456      123    4
rty678      250    8
tyu654      234    4
. . . . .
```

Skriv ett program som läser in alla bilar från filen till en vektor av bilar (max 100), sorterar vektorn först efter antal cylindrar och om samma antal efter effekt. Skriv sedan tillbaka de sorterade bilarna till samma fil, `Bilar.txt`.

15(5p) Binärfilen `Messvalues.dat` innehåller ett antal reella mätvärdesgrupper av typen :

```
struct gruppmat
{
    int gruppnr;           // Grupptillhörighet
    float x;              // Mätvärde
} gm;
```

Structarna som innehåller mätvärdena finns blandade lite hur som helst i filen. Skriv ett program som läser alla mätvärden från filen, summerar dessa till rätt grupp och beräknar och skriver ut gruppernas medelvärden.

Lösningar till tentamen i Programmering , 051119

- 1) `x = y / z + 2 * sqrt(2 + pow(y, 2));`
- 2) a) 2 b) 2
- 3) `struct bil b = {"ESA124", 145, 36.5};`
- 4) x har slutvärdet -1
- 5) `float x = 0.0;`
`char str[4] = "5.8"`

`x = str[0] - '0' + 0.1 * (str[2] - '0');`
- 6) `int trippel(int ptal)`
`{`
 `int tripptal_1, tripptal_2, tripptal_3;`

 `tripptal_1 = ptal / 10000;`
 `tripptal_2 = ptal % 10000 / 100;`
 `tripptal_3 = ptal % 100;`

 `return tripptal_1 == tripptal_2 && tripptal_1 == tripptal_3;`
`}`
- 7) `void visa_bil(struct bil b)`
`{`
 `printf("Registreringsnummer : %s\n", b.regnr);`
 `printf("Antal cylindrar : %d\n", b.nr_cyl);`
 `printf("Effekt : %.1f\n", b.effekt);`
`}`
- 8) `int is_ok(char *regnr)`
`{`
 `for (i = 0; i < 3 ; i++)`
 `{`
 `if (regnr[i] < 'A' || regnr[i] > 'Z':`
 `return false;`
 `}`

 `for (i = 3; i < 6 ; i++)`
 `{`
 `if (regnr[i] < '0' || regnr[i] > '9':`
 `return false`
 `}`
 `return true;`
`}`
- 9) `int besiktning(char *regnr)`
`{`
 `if(regnr[5] >= '0' && regnr[5] <= '5')`
 `return regnr[5] - '0' + 1;`
 `else`
 `return regnr[5] - '0' + 2;`
`}`

```

10) float str_to_float(char *str)
{
    int heldel = 0, i, k;
    float decdel = 0.0;

    for (i = 0; str[i] != '\0' && str[i] != '.'; i++)
    {
        heldel = heldel * 10 + str[i] - '0';
    }
    if (str[i] == '.')
    {
        for (k = i + 1; str[k] != '\0'; k++)
        {
            decdel = decdel * 0.1 + str[k] - '0';
        }
    }
    return heldel + decdel;
}

11) void main()
{
    int forsta, sista, tal, sum = 0;

    printf("Ge första talet : ");
    scanf("%d", &forsta);
    printf("Ge sista talet : ");
    scanf("%d", &sista);
    if (forsta < sista)
    {
        for (tal = forsta ; tal <= sista; tal++)
        {
            printf("%d + ", tal);
            sum += tal;
        }
    }
    else
    {
        for (tal = sista ; tal <= forsta; tal++)
        {
            printf("%d + ", tal);
            sum += tal;
        }
    }
    printf("\nb = %d", sum);
}

12) void main()
{
    int i, nr;
    float x, sum = 0.0, medel, rv[10];

    printf("Ge antal mätvärden (max 10, avsluta med 0) : ");
    scanf("%d", &nr);
    while (nr >= 0)
    {
        for(i = 0; i < nr; i++)
        {
            printf("Ge mätvärde %d : ", i+1);
            scanf("%f", &x);
            sum += x;
        }
        medel = sum / nr;
        for (i = 0; i < nr; i++)

```

```

    {
        if (rv[i] < 0.95 * medel || rv[i] > 1.05 * medel);
        {
            printf("%f ", rv[i]);
        }
    }
    printf("Ge antal mätvärden (max 10, avsluta med 0) : ");
    scanf("%d", &nr);
}
}

```

13) `void main()`

```

{
    char regnr[20], *month[] = {"", "januari", "februari", "mars",
                                "april", "maj", "juni", "juli",
                                "augusti", "september", "oktober",
                                "november", "december"};

    printf("Ge registreringsnummer : ");
    gets(regnr);
    while (!is_ok(regnr))
    {
        printf("Felaktigt registreringsnummer!\n");
        printf("Ge registreringsnummer : ");
        gets(regnr);
    }

    printf("Bilen ska besiktigas i %s!\n", month[besiktning(regnr)]);
}

```

14) `void main()`

```

{
    FILE *tsin;
    char filnamn[20];
    struct bil
    {
        char regnr[7];
        float effekt;
        int nr_cyl;
    } b[100], temp;
    int i, k, nr = 0;

    tsin = fopen("Bilar.txt", "r+t");

    i = 0;
    while (fscanf(tsin, "%s%d%f", b[i].regnr, &b[i].effekt,
                  &b[i].nr_cyl) != EOF)
    {
        i++;
    }
    nr = i;
    for (i = 0; i < nr - 1; i++)
    {
        for (k = i+1; k < nr; k++)
        {
            if (b[k].nr_cyl < b[i].nr_cyl)
            {
                temp = b[i];
                b[i] = b[k];
                b[k] = temp;
            }
        }
    }
}

```

```

        else if (b[k].nr_cyl == b[i].nr_cyl &&
                b[k].effekt < b[i].effekt)
        {
            temp = b[i];
            b[i] = b[k];
            b[k] = temp;
        }
    }
    rewind(tsin);

    for (i = 0; i < nr; i++)
    {
        fprintf(tsin, "%s %f %d\n", b[i].regnr, b[i].effekt,
                b[i].nr_cyl);
    }
}
fclose(tsin);
}

```

15) `void main()`

```

{
    struct gruppmat
    {
        int gruppnr;
        float x;
    }gm;

    float gruppsumma[10] = { 0.0, 0.0, 0.0, 0.0, 0.0,
                           0.0, 0.0, 0.0, 0.0, 0.0 };
    FILE *bsin;
    int gruppantal[10] = { 0, 0, 0, 0, 0,
                          0, 0, 0, 0, 0 };

    bsin = fopen("Messvalue.dat", "rb");

    fread(&gm, sizeof(struct gruppmat), 1, bsin);
    while (!feof(bsin))
    {
        gruppsumma[gm.gruppnr] += gm.x;
        gruppantal[gm.gruppnr] ++;
        fread(&gm, sizeof(struct gruppmat), 1, bsin);
    }
    for (i = 0; i < 10; i++)
    {
        if (gruppantal[i] > 0)
            printf("Medelvärden\nGruppnr%d", gruppsumma[i]/gruppantal[i]);
    }
}

```