

Lösningförslag till datorövning 7, Programmering C.

```
/* Uppg7a.c */

#include <time.h>
#include <stdlib.h>

void slump(int v[], int nr, int min, int max)
{
    int i;

    randomize();
    for (i = 0; i < nr; i++)
        v[i] = random(max - min + 1) + min;
}

int linsearch(int v[], int nr, int key)
{
    int i, svar;

    i = 0;
    while (i < nr && v[i] != key)
        i++;

    if (v[i] == key)
        svar = i + 1;
    else
        svar = 0;

    return svar;
}

#include <stdio.h>

void skriv(int v[], int nr)
{
    int i;

    for (i = 0; i < nr; i++)
    {
        if (i % 10 == 0)
            printf("\n");
        printf("%d ", v[i]);
    }
}

void main()
{
    int antal, vek[100], nyckel, svar;

    printf("Antal tal (max 100): ");
    scanf("%d", &antal);
    slump(vek, antal, 100, 999);
    skriv(vek, antal);
    printf("\n\nNyckel : ");
    scanf("%d", &nyckel);
    svar = linsearch(vek, antal, nyckel);
    if (svar)
        printf("Nyckel finns som element nr %d i vektorn", svar);
}
```

```

    else
        printf("Nyckel finns ej i vektorn");
}

/* Uppg7b.c */

#include <time.h>
#include <stdlib.h>

void slumpsort(int v[], int nr, int min, int max)
{
    int i, j, klar, tal;

    randomize();

    v[0] = random(max - min + 1) + min;
    for (i = 1; i < nr; i++)
    {
        klar = 0;
        j = i - 1;
        tal = random(max - min + 1) + min;
        do
        {
            if (tal < v[j])
            {
                v[j+1] = v[j];
                j--;
            }
            else
            {
                v[j+1] = tal;
                klar = 1;
            }
        }
        while (!klar && j > -1);

        if (!klar)
            v[0] = tal;
    }
}

int binsearch(int v[], int nr, int key)
{
    int mitti, mini = 0, maxi = nr - 1, svar;

    if (key < v[mini] || nr <= 0 || key > v[maxi])
        svar = 0;
    else
    {
        do
        {
            mitti = (mini + maxi) / 2;
            if (key < v[mitti] )
                maxi = mitti - 1;
            else if (key > v[mitti])
                mini = mitti + 1;
        }
        while (v[mittit] != key && mini <= maxi);
    }
}

```

```

        if (v[mitti] == key)
            svar = mitti + 1;
        else
            svar = 0;
    }

    return svar;
}

#include <stdio.h>

void skriv(int v[], int nr)
{
    int i;

    for (i = 0; i < nr; i++)
    {
        if (i % 10 == 0)
            printf("\n");
        printf("%d ", v[i]);
    }
}

void main()
{
    int antal, vek[100], nyckel, svar;

    printf("Antal tal (max 100) : ");
    scanf("%d", &antal);
    slumpsort(vek, antal, 1000, 9999);
    skriv(vek, antal);
    printf("\n\nNyckel : ");
    scanf("%d", &nyckel);
    svar = binsearch(vek, antal, nyckel);
    if (svar)
        printf("Nyckel finns som element nr %d i vektorn", svar);
    else
        printf("Nyckel finns ej i vektorn");
}

/* Uppg7c.c */

#include <time.h>
#include <stdlib.h>

void slump(int v[], int nr, int min, int max)
{
    int i;

    randomize();
    for (i = 0; i < nr; i++)
        v[i] = random(max - min + 1) + min;
}

void ursort(int v[], int nr)
{
    int i, j, minelemi, temp;

```

```

for (i = 0; i < nr - 1; i++)
{
    minelemi = i;
    for (j = i + 1; j < nr; j++)
    {
        if ( v[j] < v[minelemi])
            minelemi = j;
    }
    temp = v[i];
    v[i] = v[minelemi];
    v[minelemi] = temp;
}
}

#include <stdio.h>

void skriv(int v[], int nr)
{
    int i;

    for (i = 0; i < nr; i++)
    {
        if (i % 10 == 0)
            printf("\n");
        printf("%d ", v[i]);
    }
}

void main()
{
    int antal, vek[100];

    printf("Antal tal (max 100) : ");
    scanf("%d", &antal);
    slump(vek, antal, 100, 999);
    printf("\n\nOsorterade:\n");
    skriv(vek, antal);
    ursort(vek, antal);
    printf("\n\nSorterade:\n");
    skriv(vek, antal);
}

/* Uppg7d.c */

#include <time.h>
#include <stdlib.h>

void slump(int v[], int nr, int min, int max)
{
    int i;

    randomize();
    for (i = 0; i < nr; i++)
        v[i] = random(max - min + 1) + min;
}

void ursort(int v[], int nr)
{

```

```

int i, j, minelemi, temp;

for (i = 0; i < nr - 1; i++)
{
    minelemi = i;
    for (j = i + 1; j < nr; j++)
    {
        if ( v[j] < v[minelemi])
            minelemi = j;
    }
    temp = v[i];
    v[i] = v[minelemi];
    v[minelemi] = temp;
}
}

#include <stdio.h>

void skriv(int v[], int nr)
{
    int i;

    for (i = 0; i < nr; i++)
    {
        if (i % 10 == 0)
            printf("\n");
        printf("%d ", v[i]);
    }
}

void main()
{
    int antal, vek[3000];
    clock_t start, slut;

    printf("Antal tal (max 3000) : ");
    scanf("%d", &antal);
    slump(vek, antal, 100, 999);
    printf("\n\nOsorterade:\n");
    skriv(vek, antal);
    start = clock();
    ursort(vek, antal);
    slut = clock();
    printf("\n\nSorterade:\n");
    skriv(vek, antal);
    printf("\n\nSorteringstid : %f sek\n", (slut - start) / CLK_TCK);
}

/* Uppg7e.c */

#include <time.h>
#include <stdlib.h>

void slump(int v[], int nr, int min, int max)
{
    int i;

    randomize();
    for (i = 0; i < nr; i++)

```

```

        v[i] = random(max - min + 1) + min;
    }

void ursort(int v[], int nr)
{
    int i, j, minelemi, temp;

    for (i = 0; i < nr - 1; i++)
    {
        minelemi = i;
        for (j = i + 1; j < nr; j++)
        {
            if ( v[j] < v[minelemi])
                minelemi = j;
        }
        temp = v[i];
        v[i] = v[minelemi];
        v[minelemi] = temp;
    }
}

void bubbsort(int v[], int nr)
{
    int i = 0, maxi = nr - 1, bubbel = 1, temp;

    while ( bubbel && maxi > 0)
    {
        bubbel = 0;
        for (i = 0; i < maxi; i++)
        {
            if ( v[i] > v[i + 1] )
            {
                temp = v[i];
                v[i] = v[i + 1];
                v[i + 1] = temp;
                bubbel = 1;
            }
        }
        maxi--;
    }
}

void insort(int v[], int s[], int nr)
{
    int i, j, klar;

    s[0] = v[0];
    for (i = 1; i < nr; i++)
    {
        klar = 0;
        j = i - 1;
        do
        {
            if (v[i] < s[j])
            {
                s[j+1] = s[j];
                j--;
            }
            else
            {
                s[j+1] = v[i];
            }
        }
    }
}

```

```

        klar = 1;
    }
}
while (!klar && j > -1);

    if (!klar)
        s[0] = v[i];
}
}

#include <stdio.h>
#define antal 2000

void main()
{
    int vek[antal], svek[antal];
    clock_t start, slut;

    slump(vek, antal, 100, 999);
    start = clock();
    ursort(vek, antal);
    slut = clock();
    printf("\n\nSorteringstid ursort : %f sek\n", (slut - start) / CLK_TCK);

    slump(vek, antal, 100, 999);
    start = clock();
    bubbsort(vek, antal);
    slut = clock();
    printf("\n\nSorteringstid bubbsort : %f sek\n", (slut - start)/CLK_TCK);

    slump(vek, antal, 100, 999);
    start = clock();
    insort(vek, svek, antal);
    slut = clock();
    printf("\n\nSorteringstid insort : %f sek\n", (slut - start) / CLK_TCK);
}

```

```
/* Uppg7f.c */
```

```
#include <string.h>
```

```
int linsearch_str(char *v[], int nr, char *key)
{
    int i, svar;

    i = 0;
    while (i < nr && strcmp(v[i], key) != 0)
        i++;

    if (strcmp(v[i], key) == 0)
        svar = i + 1;
    else
        svar = 0;

    return svar;
}

```

```
int binsearch_str(char *v[], int nr, char *key)
{

```

```

int mitti, mini = 0, maxi = nr - 1, svar;

if (strcmp(key, v[mini]) < 0 || nr <= 0 || strcmp(key, v[maxi]) > 0)
    svar = 0;
else
{
    do
    {
        mitti = (mini + maxi) / 2;
        if (strcmp(key, v[mitti]) < 0)
            maxi = mitti - 1;
        else if (strcmp(key, v[mitti]) > 0)
            mini = mitti + 1;
    }
    while (strcmp(v[mitti], key) != 0 && mini <= maxi);

    if (strcmp(v[mitti], key) == 0)
        svar = mitti + 1;
    else
        svar = 0;
}
return svar;
}

#include <stdio.h>

void main()
{
    char *strvek1[10] = {"Ulrika", "Kalle", "Ola", "Berit", "Eva",
                       "Per", "Rut", "Olle", "Birgitta", "Rolf"};

    /*Sorterad strängvektor för binär sökning*/
    char *strvek2[10] = {"Berit", "Birgitta", "Eva", "Kalle", "Ola",
                       "Olle", "Per", "Rolf", "Rut", "Ulrika"};

    char namn[20];
    int svar;

    /*Linjär sökning*/
    printf("Vem söks? ");
    gets(namn);
    svar = linsearch_str(strvek1, 10, namn);
    if (svar)
        printf("Namnet finns på plats nr %d\n", svar);
    else
        printf("Namnet finns ej!\n");

    /*Binär sökning*/
    printf("Vem söks? ");
    gets(namn);
    svar = binsearch_str(strvek2, 10, namn);
    if (svar)
        printf("Namnet finns på plats nr %d\n", svar);
    else
        printf("Namnet finns ej!\n");
}

```



```

/* Uppg7g.c */

#include <string.h>

void ursort_str(char *v[], int nr)
{
    int i, j, minelemi;
    char *temp;

    for (i = 0; i < nr - 1; i++)
    {
        minelemi = i;
        for (j = i + 1; j < nr; j++)
        {
            if ( strcmp(v[j], v[minelemi]) < 0)
                minelemi = j;
        }
        temp = v[i];
        v[i] = v[minelemi];
        v[minelemi] = temp;
    }
}

void bubbsort_str(char *v[], int nr)
{
    int i = 0, maxi = nr - 1, bubbel = 1;
    char *temp;

    while ( bubbel && maxi > 0)
    {
        bubbel = 0;
        for (i = 0; i < maxi; i++)
        {
            if ( strcmp(v[i], v[i + 1]) > 0 )
            {
                temp = v[i];
                v[i] = v[i + 1];
                v[i + 1] = temp;
                bubbel = 1;
            }
        }
        maxi--;
    }
}

void insert_str(char *v[], char *s[], int nr)
{
    int i, j, klar;

    s[0] = v[0];
    for (i = 1; i < nr; i++)
    {
        klar = 0;
        j = i - 1;
        do
        {
            if (strcmp(v[i], s[j]) < 0)
            {
                s[j+1] = s[j];
                j--;
            }
        }
    }
}

```

```

        else
        {
            s[j+1] = v[i];
            klar = 1;
        }
    }
    while (!klar && j > -1);

    if (!klar)
        s[0] = v[i];
}
}

#include <stdio.h>

void skriv_str(char *v[], int nr)
{
    int i;

    for(i = 0; i < nr; i++)
        puts(v[i]);
}

void main()
{
    char *strvek1[5] = {"Ulrika", "Kalle", "Ola", "Berit", "Eva"};
    char *strvek2[5] = {"Per", "Rut", "Olle", "Birgitta", "Rolf"};
    char *strvek3[5] = {"Lena", "Kurt", "Lars", "Emma", "Peter"};
    char *sortvek[5];

    skriv_str(strvek1, 5);
    printf("\n");
    ursort_str(strvek1, 5);
    skriv_str(strvek1, 5);

    printf("\n");
    skriv_str(strvek2, 5);
    printf("\n");
    bubbsort_str(strvek2, 5);
    skriv_str(strvek2, 5);

    printf("\n");
    skriv_str(strvek3, 5);
    printf("\n");
    insort_str(strvek3, sortvek, 5);
    skriv_str(sortvek, 5);
}

/* Uppg7h.c */

#include <string.h>

void ursort_str(char *v[], int nr)
{
    int i, j, minelemi;
    char *temp;

    for (i = 0; i < nr - 1; i++)
    {

```

```

        minelemi = i;
        for (j = i + 1; j < nr; j++)
        {
            if ( strcmp(v[j], v[minelemi]) < 0)
                minelemi = j;
        }
        temp = v[i];
        v[i] = v[minelemi];
        v[minelemi] = temp;
    }
}

#include <stdio.h>

void main()
{
    char namn[20][30], *strvek[20];
    FILE *tsinut;
    int i, nr = 0;

    /*Läs namn från fil till vektor*/
    tsinut = fopen("namn.txt", "r+");
    while (fgets(namn[nr], 30, tsinut) != NULL)
    {
        strvek[nr] = namn[nr];
        nr++;
    }

    ursort_str(strvek, nr);

    /*Skriv de sorterade namnen tillbaka till fil*/

    fprintf(tsinut, "\n");
    for (i = 0; i < nr; i++)
    {
        fputs(strvek[i], tsinut);
    }
}

```