

Heimadæmi 3 - Tölvutækni og forritun

Magnús Daníel Einarsson

September 2023

Verkefni 1

Ef p og q eru bendar á heiltölur (þ.e. int^*) getur þá $(p == q)$ verið satt án þess að $(*p == *q)$ sé satt? En öfugt (þ.e. $(*p == *q)$ er satt, en $(p == q)$ er ósatt)? Rökstyðjið svör ykkar í báðum tilvikum.

Lausn:

$*p == *q$ er bendir á innihald í hólfi á meðan að $p == q$ er bendir á minnissvæði sjálft. Segjum að $*p = 6$ þá er $*q$ líka 6. ef $p == q$ er með sama minnissvæði þá verður innihaldið að vera það sama til þess að dæmið gangi upp. Segjum að minnishólfið sé $0x0AF428$ fyrir p þá er q með sama minnishólf. Innihaldið verður því að vera það sama. Þess vegna er fyrra dæmið vitlaust því $*p$ verður að vera með sama innihald og $*q$ ef p og q er það sama. Hinsvegar getur seinna dæmið gengið upp. Ef við erum með $*p == *q$ og segjum að báðar tölur eru 5 og svo erum við með minnissvæði $0xAF0164$ og minnissvæði $0xAB6012$ þá geta bæði minnissvæði verið með sömu heiltölu en samt bent á mismunandi minnissvæði.

Verkefni 2

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

int *get_ints(size_t n)
{
    int *ip;
    ip = (int *)calloc(n, sizeof(int));

    if (ip == NULL)
    {
        printf("Can't allocate %zu ints\n", n);
    }
    return ip;
}

int main(int argc, char **argv)
{
    srand(time(NULL));
    if (argc < 3)
    {
        printf("Usage: %s <filename>\n", argv[0]);
        return 1;
    }
    int n = atoi(argv[1]);
    int k = atoi(argv[2]);
    int *ip = get_ints(n);
    for (int i = 0; i < n * k; i++)
    {
        int r = rand() % (n - 1);
        ip[r] += 1;
    }
    int cnt = 0;
    for (int j = 0; j < n; j++)
    {
        if (ip[j] == 0)
            cnt += 1;
    }

    float percent = ((float)cnt / n) * 100;
    printf("Eftir %d ítranir eru %d hólf ennpá 0, eða %.2f %%\n", n * k, cnt, percent);
    free(ip);
    return 0;
}
```

```
● (base) magnusde93@tg-dw072 ~/University-Assignments/Semester-3/Com
Eftir 2000 ítranir eru 11 hólf ennpá 0, eða 2.20 %
```

Verkefni 3

```
void delNode(struct dNode **h, struct dNode **t, int k)
{
    struct dNode *p = *h;
    // eyða fyrsta hnúti
    if (k == 1)
    {
        p = p->next;
        free(*h);
        *h = p;
    }
    // leita að og eyða hnút öðrum en fyrsta
    else
    {
        // leitum að staki númer k
        for (int i = 1; i < k - 1 && p->next != NULL; i++)
        {
            p = p->next;
        }
        if (p->next == NULL)
        {
            return;
        }

        struct dNode *temp = p->next;
        p->next = temp->next;
        free(temp);
    }
    // skilum bendi á fyrsta hnútinn
    return;
}
```

```
● (base) magnusde93@tg-dw072 ~/University-Assigments/Semester-3/Comp
Listi: 7 49 73 58 30 72 44 78 23 9
Listi: 49 73 58 30 72 44 78 23 9
Listi: 49 73 30 72 44 78 23 9
Listi: 49 73 30 72 44 78 23 9
```

Verkefni 5

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

#define MAX_LINE_LENGTH 1024
#define MAX_FIELDS 50

struct realestate
{
    int total;
    int size_of_post_code;
};

int main()
{
    int sum = 0;
    int population = 0;
    char line[MAX_LINE_LENGTH];
    fgets(line, sizeof(line), stdin);

    char *fields[MAX_FIELDS];
    struct realestate *ip = calloc(999, sizeof(struct realestate));
    while (fgets(line, sizeof(line), stdin))
    {
        line[strcspn(line, "\n")] = '\0';

        char *token = strtok(line, "\t;");

        int numFields = 0;
        while (token != NULL)
        {
            if (numFields < MAX_FIELDS)
            {
                fields[numFields] = token;
                numFields++;
            }
            else
            {
                fprintf(stderr, "Too many fields in the CSV line.\n");
                exit(1);
            }

            token = strtok(NULL, "\t;");
        }

        if (atoi(fields[5]) < 999)
        {
            ip[atoi(fields[5])].total += atoi(fields[11]);
            ip[atoi(fields[5])].size_of_post_code += 1;
            sum += atoi(fields[11]);
        }
        else
        {

```

```

        sum += atoi(fields[10]);
    }
    population += 1;
}
for (int i = 0; i < 901; i++)
    if (ip[i].total != 0)
    {
        printf("%d %d %d %f %f\n", i, ip[i].total, ip[i].size_of_post_code, ((float)ip[i].total / population), ((float)ip[i].size_of_post_code / population));
    }

free(ip);
return 0;
}

```

```

● (base) ✖ magnusde93@tg-dw072 ~/University-Assignments/Semester-3/Co
101 109005627 638 0.095221 0.049957
102 11013726 119 0.009621 0.009318
103 8936596 98 0.007807 0.007674
104 42173537 407 0.036840 0.031869
105 67783923 649 0.059212 0.050818
107 24859785 237 0.021716 0.018558
108 45648442 414 0.039876 0.032417
109 18812828 270 0.016434 0.021142
110 88959196 442 0.077710 0.034610
111 12764631 239 0.011150 0.018714
112 29707281 401 0.025951 0.031399
113 16537513 219 0.014446 0.017148
116 642900 13 0.000562 0.001018
162 1618500 12 0.001414 0.000940
170 7622004 87 0.006660 0.006012

```