

Heimadæmi 3
TÖL309G Tölvutækni og forritun
Hjörvar Sigurðsson

1.

1.

Minni:

Address	Value	
0x1000	0x3	int p = 3
0x1004	0x3	int q = 3
0x1008	0x1000	int *p = &p
0x100C	0x1004	int *q = &q
~		
~		
~		

int *p = &p

int *q = &q

• Segjum að int p = 3 • Segjum að int q = 3

i)

Hér er $p == q$, þar sem bæði p og q
 jafngilda 3, en $p*$ jafngildir
 ekki $q*$, þar sem heiltölurnar
 p og q eru geymdar í
 sitt hvora minnishélf; en $p*$
 og $q*$ benda því á sitt hvort
 minnishélf.

Svarið er því: Já, það getur verið.

ii)

$p* == q*$ þýðir að bendarnir
 benda á sama minnishélf.
 Það er ekki hægt að bendarnir
 bendi á sama minnishélf
 en að tveir mismunandi
 heiltölur séu geymdar þar.

Því: ef $p* == q*$, þá $p == q$.

Semsagt:

- i) Ef að $p = 3$, og $q = 3$, þá er $p == q$, en p^* jafngildir ekki q^* þar sem heiltölurnar p og q eru geymdar í sitthvorum minnishólfi en p^* og q^* benda því á sitthvort hólfið. Svarið við fyrri spurningunni er því: Já, það getur verið.
- ii) $p^* == q^*$ þýðir að bendarnir benda á sama minnishólfið. Það er ekki hægt að bendarnir bendi á sama minnishólfið en að tvær mismunandi heiltölur séu geymdar þar. Því: ef $p^* == q^*$, þá $p == q$.

2.

Kóði:

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

int main(int argc, char *argv[]) {

    // Skipun til að fá mismunandi slembirunu í hvert sinn.
    srand(time(0));

    // Athugar hvort inntak er rétt.
    if (argc < 2) {
        exit(1);
    }

    // Upphafstillir breytur með inntaki.
    int n = atoi(argv[1]);
    int k = atoi(argv[2]);

    // Debug prentun.
    printf("n = %d, k = %d.\n", n, k);

    // Býr til n-staka heiltölufylkið a með calloc.
    int *a = (int*)calloc(n, sizeof(int));

    // Bý til slembitölu r og hækka a[r] um einn; endurtek n*k sinnum.
    for (int i = 0; i < (k * n); i++) {
        int r = rand() % n;
        a[r] += 1;
    }

    // Athuga í hve mörgum hólfum í a er 0.
    int count = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] == 0) {
            count += 1;
        }
    }
}
```

```

    }

    // Skila í hve mörgum hólfum a er 0.
    int itr = n * k;
    float hlutfall = ((float)count / (float)n) * 100;
    printf("Eftir %d itranir eru %d holf ennthá 0, eda %.1f%%.", itr, count,
hlutfall);

    return 0;
}

```

Keyrsla:

```

aemi-3> & .\"slembi.exe" 500 4
n = 500, k = 4.
Eftir 2000 itranir eru 9 holf ennthá 0, eda 1.8%.
aemi-3> & .\"slembi.exe" 1000 5
n = 1000, k = 5.
Eftir 5000 itranir eru 7 holf ennthá 0, eda 0.7%.

```

3. Kóði:

```

/*****
    Beinagrind fyrir dæmi 3 í heimadæmum 3 í Tölvutækni og forritun,
    haust 2022

    Sjá lýsingu verkefnis á dæmablaði
    *****/
#include <stdio.h>
#include <stdlib.h>

struct dNode {
    int data;
    struct dNode *prev;
    struct dNode *next;
};

/* Prentar út stök í tvítengdum list */
void printList(struct dNode *h) {
    printf("Listi: ");
    while (h != NULL) {
        printf("%d ", h->data);
        h = h->next;
    }
    printf("\n");
}

```

```

/* Eyðir hnúti númer k í tvítengda listanum sem h og t benda á.
   Breytir mögulega haus- og halabendunum h og t */
void delNode(struct dNode *h, struct dNode *t, int k) {

    // Ef eyða á fyrsta hnút.
    if (k == 1) {
        *h = *h->next;
    }

    struct dNode *temp = h;
    // Annars:
    for (int i = 0; i < k-1; i++) {
        temp = temp->next;
    }
    struct dNode *temp2 = temp->prev;
    temp2->next = temp->next;
    temp->next->prev = temp2;

    //printf("debug: %d\n", temp->data);
}

/* Býr til n-staka lista með slembigildum (0 til 99),
   skilar bendi á fremsta hnút, eða NULL ef n < 1 */
void createList(struct dNode **h, struct dNode **t, int n) {
    struct dNode *p;
    int i;

    /* Ef n er núll eða minna þá tómur listi */
    if (n < 1) {
        *h = NULL;
        *t = NULL;
        return;
    }

    /* Búa til fyrsta hnútinn og láta haus og hala benda á hann */
    *h = *t = p = (struct dNode *)malloc(sizeof(struct dNode));
    (*h)->data = rand()%100;
    (*h)->prev = NULL;
    (*h)->next = NULL;

    /* Búa til restina af hnútunum og setjum þá aftast */
    for(i=1; i<n; i++) {
        p = (struct dNode *)malloc(sizeof(struct dNode));
        p->data = rand()%100;
        p->prev = *t;
        p->next = NULL;
        (*t)->next = p;
    }
}

```

```

        *t = p;
    }
}

int main() {
    int i;

    struct dNode *head = NULL;
    struct dNode *tail = NULL;

    /* Búa til listann með 10 slembigildum */
    createList(&head, &tail, 10);
    printList(head);

    /* Eyða út nokkrum hnútum og skoða listann í hvert sinn */
    //printf("Debug - a ad eyda hnut 1, eda 41.\n");
    delNode(head, tail, 1);
    printList(head);
    //printf("\n");

    //printf("Debug - a ad eyda hnut 3, eda 0.\n");
    delNode(head, tail, 3);
    printList(head);
    //printf("\n");

    //printf("Debug - a ad eyda hnut 10, en thad fer umfram listann.\n");
    delNode(head, tail, 10);
    printList(head);
    //printf("\n");

    return 0;
}

```

Keyrsla:

```

> & .\"dllist.exe"
Listi: 41 67 34 0 69 24 78 58 62 64
Listi: 67 34 0 69 24 78 58 62 64
Listi: 67 34 69 24 78 58 62 64

```

4. Kóði:

```

/*****
Beinagrind að lausn á dæmi 4 í heimadæmum 3 í Tölvutækni
og forritun, haust 2022

```

```

    Sjá lýsingu á dæmablaði
    *****/
#include <stdio.h>
#include <string.h>

#define MAX_LEN 200

int main(int argc, char **argv) {
    char innstr[MAX_LEN];

    while (scanf("%s", innstr) != EOF) {
        if (strstr(innstr, "x") != NULL) {
            int l = strlen(innstr);
            for (int i = 0; i < l; i++) {
                printf("%s", "x");
            }
            printf("\n");
        }

        else if(strstr(innstr, "X") != NULL) {
            int l = strlen(innstr);
            for (int i = 0; i < l; i++) {
                printf("%s", "x");
            }
            printf("\n");
        }

        else {
            printf("%s\n", innstr);
        }
    }

    return 0;
}

```

Keyrsla:

```
aemi-3> & .\"xsia.exe"  
Hallo  
Hallo  
Eg  
Eg  
heiti  
heiti  
Xavier  
xxxxxx  
abcdefg  
abcdefg  
xyz  
xxx  
abcdefghijklradkfj  
xxxxxxxxxxxxxxxxxxxxxx
```