

Programski jezik I

– Primjeri sa 8. termina predavanja –

Dinamička alokacija memorije

1. Napisati program kojim se unosi niz x dužine n . Program treba da kreira niz y od elementa niza x koji su manji od elmenta nakon sebe. Memoriju za nizove x i y zauzeti dinamički.

Primjer: Ako se unese niz $x = [3 \ 6 \ 5 \ 8 \ 1]$, program treba da kreira i odštampa niz $y = [3 \ 5]$.

Rješenje:

```
01 | #include <stdio.h>
02 | #include <stdlib.h>
03 |
04 | int main()
05 | {
06 |     int *x, *y;
07 |     int n;
08 |     printf("Unijeti dužinu niza");
09 |     scanf("%d", &n);
10 |     x = (int *)malloc(n * sizeof(int));
11 |     if (x == NULL)
12 |     {
13 |         printf("Neuspjela alokacija memorije");
14 |         return 1;
15 |     }
16 |
17 |     for (int i = 0; i < n; i++)
18 |     {
19 |         printf("Unesite %d. element niza: ", i + 1);
20 |         scanf("%d", &x[i]);
21 |     }
22 |
23 |     int br = 0;
24 |     for (int i = 0; i < n - 1; i++)
25 |     {
26 |         if (x[i] < x[i + 1])
27 |         {
28 |             br++;
29 |         }
30 |     }
31 |     y = (int *)malloc(br * sizeof(int));
32 |     if (y == NULL)
33 |     {
34 |         printf("Neuspjela alokacija memorije");
35 |         return 1;
36 |     }
37 |
38 |     int j = 0;
39 |     for (int i = 0; i < n - 1; i++)
40 |     {
41 |         if (x[i] < x[i + 1])
42 |         {
43 |             y[j++] = x[i];
```

```

44 |         }
45 |     }
46 |
47 |     for (int i = 0; i < br; i++)
48 |     {
49 |         printf("%d ", y[i]);
50 |     }
51 | }

```

2. Napisati program koji sadrži funkciju `ogranici` kojoj se prosljeđuje niz cijelih brojeva `x` dužine `n` i cio broj `prag`. Funkcija treba da kreira i vrati novi niz `y` koji se sastoji od istih elemenata kao i niz `x`, pri čemu su svi elementi koji su veći od `prag` zamijenjeni sa vrijednošću `prag`.

Primjer: Ako se unese niz `x = [3 5 1 7 0]` i `prag = 3`, funkcija treba da kreira i vrati niz `x = [3 3 1 3 0]`.

Rješenje:

```

01 | #include <stdio.h>
02 | #include <stdlib.h>
03 |
04 | int* ogranici(int *, int, int);
05 | int main()
06 | {
07 |     int n = 5, y_duzina, prag = 3;
08 |     int x[] = {9, 1, 5, 2, 7};
09 |     int *y;
10 |
11 |     y = ogranici(x, n, prag);
12 |
13 |     for (int i = 0; i < n; i++)
14 |     {
15 |         printf("%d ", y[i]);
16 |     }
17 | }
18 |
19 | int* ogranici(int *x, int n, int prag)
20 | {
21 |     int *temp;
22 |     temp = (int *)malloc(n * sizeof(int));
23 |     if (temp == NULL)
24 |     {
25 |         printf("Neuspjela alokacija memorije.");
26 |         return NULL;
27 |     }
28 |     for (int i = 0; i < n; i++)
29 |     {
30 |         if (x[i] <= prag)
31 |         {
32 |             temp[i] = x[i];
33 |         }
34 |         else
35 |         {
36 |             temp[i] = prag;
37 |         }
38 |     }
39 |     return temp;
40 | }

```