

## 5\_2\_10

Lista kroków:

1. Utwórz funkcję o nazwie 'wytnij2', przyjmującą jako argumenty: dwa wskaźniki do napisów (char \*) nap1 i nap2.
2. Sprawdź, czy którykolwiek z napisów jest pusty. Jeśli tak, zakończ funkcję bez wprowadzania zmian w napisie nap1.
3. Znajdź długość napisu nap1 i nap2, iterując przez nie, aż do napotkania znaku zerowy ('\0').
4. Używając pętli, przeszukaj napis nap1, aby znaleźć początek pierwszego wystąpienia napisu nap2 w nap1.
5. Jeśli nap2 występuje w nap1, usuń nap2 z nap1, przesuując znaki nap1 na indeksach większych niż pozycja+długosc\_nap2 do indeksów większych niż pozycja, jednocześnie uaktualniając nap1.
6. Ustaw znak zerowy ('\0') na odpowiedniej pozycji w nap1, aby zakończyć go po przesunięciu znaków.

Kod:

```
#include <stdio.h>

void wytnij2(char *nap1, char *nap2) {
    if (nap1[0] == '\0' || nap2[0] == '\0') {
        return;
    }

    int i, j, dlugosc_nap1 = 0, dlugosc_nap2 = 0;
    int pozycja = -1;

    while (nap1[dlugosc_nap1] != '\0') {
        dlugosc_nap1++;
    }
```

```

while (nap2[dlugosc_nap2] != '\0') {
    dlugosc_nap2++;
}

for (i = 0; i <= dlugosc_nap1 - dlugosc_nap2 && pozycja == -1; i++) {
    for (j = 0; j < dlugosc_nap2 && nap1[i + j] == nap2[j]; j++);

    if (j == dlugosc_nap2) {
        pozycja = i;
    }
}

if (pozycja != -1) {
    for (i = pozycja + dlugosc_nap2; i <= dlugosc_nap1; i++) {
        nap1[pozycja++] = nap1[i];
    }
}

}

int main() {
    char nap1[] = "To jest przykladowy napis.";
    char nap2[] = "przykladowy";

    wytnij2(nap1, nap2);
    printf("Wynik po wycieciu: %s\n", nap1);

    return 0;
}

```