

## 5\_2\_11

Lista kroków:

1. Zdefiniuj funkcję `wytnijzw` przyjmującą dwa argumenty - `nap1` i `nap2`.
2. Zdefiniuj wskaźnik `nap1_iter` do iteracji po `nap1`.
3. Zdefiniuj wskaźnik `nap1_write` do zapisu przefiltrowanych znaków do `nap1`.
4. Iteruj przez każdy znak w `nap1` (aż do napotkania znaku `'\0'`).
5. Dla każdego znaku w `nap1`, iteruj przez wszystkie znaki w `nap2`.
6. Jeżeli znak z `nap1` występuje w `nap2`, zakończ wewnętrzną iterację i przejdź do następnego znaku z `nap1`.
7. Jeżeli znak z `nap1` nie występuje w `nap2`, zapisz go pod adresem wskaźnika `nap1_write` i przesuń wskaźnik `nap1_write` o jeden.
8. Po przefiltrowaniu wszystkich znaków z `nap1`, zakończ zmodyfikowany napis `nap1` znakiem `'\0'`.
9. Zakończ funkcję.

```
#include <stdio.h>

int zawiera_znak(char *nap2, char znak) {
    for (; *nap2 != '\0'; ++nap2) {
        if (*nap2 == znak) {
            return 1;
        }
    }
    return 0;
}

void wytnijzw(char *nap1, char *nap2) {
    char *nap1_iter = nap1;
    char *nap1_write = nap1;

    for (; *nap1_iter != '\0'; ++nap1_iter) {
        if (!zawiera_znak(nap2, *nap1_iter)) {
```

```

        *nap1_write = *nap1_iter;
        ++nap1_write;
    }
}
*nap1_write = '\0';
}

int main() {
    char nap1[] = "abcdefghijk3f";
    char nap2[] = "fhj";
    wytnijzw(nap1, nap2);
    printf("Po wycieciu znakow: %s\n", nap1);
    return 0;
}

```

Inny sposób:

1. Zdefiniuj funkcję `wytnijzw` przyjmującą dwa argumenty - `nap1` i `nap2`.
2. Zdefiniuj zmienną `i` do iteracji po `nap1`.
3. Zdefiniuj zmienną `write_index` do zapisu przefiltrowanych znaków do `nap1`.
4. Iteruj przez każdy znak w `nap1` (aż do napotkania znaku `'\0'`).
5. Dla każdego znaku w `nap1`, iteruj przez wszystkie znaki w `nap2`.
6. Jeżeli znak z `nap1` występuje w `nap2`, zakończ wewnętrzną iterację i przejdź do następnego znaku z `nap1`.
7. Jeżeli znak z `nap1` nie występuje w `nap2`, zapisz go pod indeksem `write_index` w `nap1` i zwiększ wartość `write_index`.
8. Po przefiltrowaniu wszystkich znaków z `nap1`, zakończ zmodyfikowany napis `nap1` znakiem `'\0'`.
9. Zakończ funkcję.

```

#include <stdio.h>

int zawiera_znak(char *nap2, char znak) {
    for (int i = 0; nap2[i] != '\0'; ++i) {
        if (nap2[i] == znak) {
            return 1;
        }
    }
    return 0;
}

```

```

void wytnijzw(char *nap1, char *nap2) {
    int write_index = 0;

    for (int i = 0; nap1[i] != '\0'; ++i) {
        if (!zawiera_znak(nap2, nap1[i])) {
            nap1[write_index] = nap1[i];
            ++write_index;
        }
    }
    nap1[write_index] = '\0';
}

int main() {
    char nap1[] = "abcdefghijkf";
    char nap2[] = "fhj";
    wytnijzw(nap1, nap2);
    printf("Po wycieciu znakow: %s\n", nap1);
    return 0;
}

```