

1. Példa

Az alábbi C forráskód segítségével egy, a felhasználótól bekért valós számról állapíthatjuk meg, hogy kisebb-e 0-nál, 10-nél, 100-nál, vagy 1000-nél. Ebben az esetben a többirányú elágazást biztosító *if-else if-else* szerkezet nem váltható ki *switch-case* szerkezettel, mert az egyes ágak feltételeit előíró kifejezésekben nem a szám megegyezését vizsgáljuk valamilyen konstansok vagy kifejezések értékeivel.

```
#include <stdio.h>
#define NULLA 0
#define TIZ 10
#define SZAZ 100
#define EZER 1000
main(){
    float x;

    printf("\nA program meghatározza, hogy egy szám kisebb-e \
0-nal, 10-nel, 100-nal vagy 1000-nel.\n");
    printf("Adjon meg egy számot: ");
    scanf("%f",&x);
    if (x<NULLA)
        printf("\nA szám kisebb, mint 0.\n");
    else if (x<TIZ)
        printf("\nA szám kisebb, mint 10.\n");
    else if (x<SZAZ)
        printf("\nA szám kisebb, mint 100.\n");
    else if (x<1000)
        printf("\nA szám kisebb, mint 1000.\n");
    else
        printf("\nA szám nagyobb, vagy egyenlő 1000-el.\n");
}
```

2. Példa

Az alábbi C forráskód segítségével két, a felhasználótól bekért valós szám közötti alpműveleteket végezhetünk el. Az alpműveletek jelei: + - * /

A bemenet megadásának formája egy szorzási művelet példáját alapul véve: 2.15*6.02

A kód gondoskodik a számok és a számokat elválasztó karakter beolvasásáról. Megvizsgálja a karaktert, és egyezést keres az alpműveletek karaktereivel. Amennyiben megtalálja az egyezést, az annak megfelelő műveletet hajtja végre a két szám között, és a végeredményt kiírja a képernyőre. Ha nem talál egyezést, vagy a beolvasás során hiba következett be, akkor hibüzenet megjelenítése után leáll. Az egyezések keresése ebben a kódban többirányú elágazást biztosító if-else if-else szerkezettel van megvalósítva. A 3. példában ugyanez a feladat switch-case szerkezettel van megoldva. Mivel egyezést keresünk a beolvasott karakter és az alpműveletek karakterei között, mindkét szerkezet alkalmazható.

```
#include <stdio.h>
main(){
    float a, b, c;
    char op;

    printf("\nA program kiszámítja az eredményt két szám között \
kiválasztott alpműveletnek.\n");
    printf("Kérem gépelje be a két számot elválasztva a kívánt \
műveleti jellel (+ - * /):\n ");
    if (scanf("%f%c%f",&a,&op,&b)==3){
        if (op=='+')
            c=a+b;
        else if (op=='-')
            c=a-b;
        else if (op=='*')
            c=a*b;
        else if (op=='/')
            c=a/b;
        else{
            printf("\nA művelet nem hajtható végre. \
A program leáll.\n");
            return;
        }
        printf("\n%f%c%f= %f\n",a,op,b,c);
    }
    else{
        printf("\nA beolvasási művelet sikertelen. \
A program leáll.\n");
        return;
    }
}
```

```

A program kiszámítja az eredményt két szám között kiválasztott alapműveletnek.
Kérem gépelje be a két számot elválasztva a kívánt műveleti jellel (+ - * /):
-4.25*3.14
-4.250000*3.140000 = -13.345000
-----
Process exited after 51.97 seconds with return value 33
Press any key to continue . . . █

```

3. Példa

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

```

main(){
    float a, b, c;
    char op;

    printf("\nA program kiszámítja az eredményt két szám között \
kiválasztott alapműveletnek.\n");
    printf("Kérem gépelje be a két számot elválasztva a kívánt \
műveleti jellel (+ - * /):\n ");
    if (scanf("%f%c%f",&a,&op,&b)==3){
        switch(op){
            case '+':
                c=a+b;
                break;
            case '-':
                c=a-b;
                break;
            case '*':
                c=a*b;
                break;
            case '/':
                c=a/b;
                break;
            default:{
                printf("\nA művelet nem hajtható végre. A program \
leáll.\n");
                return;
            }
        }
        printf("\n%f%c%f = %f\n",a,op,b,c);
    }
    else{
        printf("\nA beolvasási művelet sikertelen. A program \
leáll.\n");
        return;
    }
}

```