# טיפוסי משתנים בשפת C

**משפחת השלמים**

נהוג להגדיר משתנה מטיפוס שלם, אם ידוע לנו כי המשתנה מייצג נתונים שלמים בלבד, כגון : מספר אנשים, מספר פריטים וכו’.

|  |  |  |
| --- | --- | --- |
| **שם הטיפוס** | **גודל ב – BYTES** | **תחום ההצבה האפשרי** |
| short int | 2 | -32,768 to +32,767 |
| int | 4 | -2,147,483,648 to +2,147,483,647 |
| long | 4 | -2,147,483,648 to +2,147,483,647 |
| unsigned int | 2 | 0 to 65,535 |
| unsigned long | 4 | 0 to 4,294,967,295 |

**משפחת הממשיים**

נהוג להגדיר משתנה מטיפוס ממשי, אם ידוע לנו כי המשתנה מייצג נתונים העשויים להכיל שברים, או עשויים להיות גדולים מתחום ההגדרה של שלמים.

|  |  |  |
| --- | --- | --- |
| **שם הטיפוס** | **גודל ב – BYTES** | **תחום ההצבה האפשרי** |
| float | 4 | 3.4\*10-38 to 3.4\*10+38 |
| double | 8 | 1.7\*10-308 to 1.7\*10+308 |
| long double | 10 | 3.4\*10-4932 to 3.4\*10+4932 |

**משפחת התווים**

נהוג להגדיר משתנה מטיפוס תו, אם ידוע לנו כי המשתנה אמור להכיל תווים כלשהם (אותיות, ספרות, סימנים וכו').

|  |  |  |
| --- | --- | --- |
| **שם הטיפוס** | **גודל ב – BYTES** | **תחום ההצבה האפשרי** |
| Char | 1 | -128 to +127 |
| unsigned char | 1 | 0 to 255 |

# תווי בקרה לפונקציות קלט פלט בשפת C

הפונקציות scanf() ו – printf() משתמשות באותם תווי בקרה על מנת לקלוט / לפלוט נתונים. להלן רשימת תווי הבקרה הקיימים בשפת C :

**תו הבקרה מייצג טיפוס**

d% שלם בבסיס 10 char / int

hd% שלם בבסיס 10 short int

ld% שלם בבסיס 10 long int

u% שלם בבסיס 10 unsigned int

lu% שלם בבסיס 10 unsigned long

o% שלם בבסיס 8 char / int

x% שלם בבסיס 16 char / int

c% תו בודד char / unsigned char

s% מחרוזת [?]? char

f / %e% ממשי בתצוגה מדעית/עשרונית float

lf / %le% ממשי בתצוגה מדעית/עשרונית double

g% הקצר מבין מדעית/עשרונית float / double

# טבלת אופרטורים מסכמת – סדר קדימויות

**Precedence of Operators**

בטבלה הבאה מחולקים כל האופרטורים בשפת C ל – 14 קטגוריות.

הקטגוריה הראשונה בסדר קדימות עליון, והאחרונה בעלת סדר קדימות אחרון.

האופרטורים בכל קבוצה הם שווי ערך מבחינת סדר קדימות.

|  |  |  |  |
| --- | --- | --- | --- |
| הסבר האופרטור | **האופרטור** | **שם הקטגוריה** | **מס'** |
| Function call | () | Highest | 1 |
| Array subscript | [] |
| indirect component selector | -> |
| direct component selector | . |
| Logical NOT | ! | Unary | 2 |
| Bitwise (1's) complement | ~ |
| Unary minus | - |
| Preincrement or postincrement | ++ |
| Predecrement or postdecrement | -- |
| Address | & |
| Indirection (pointer) | \* |
| Multiply | \* | Multiplicative | 3 |
| Divide | / |
| Remainder (modulus( | % |
| Plus | + | Additive | 4 |
| minus | - |
| Shift left | << | Shift | 5 |
| Shift right | >> |
| Less than | < | Relational | 6 |
| Less than or equal to | >= |
| Greater than | > |
| Greater than or equal to | <= |
| Equal to | == | Equality | 7 |
| Not equal to | != |
| Bitwise AND | & |  | 8 |
| Bitwise XOR | ^ |  | 9 |
| Bitwise OR | | |  | 10 |
| Logical AND | && |  | 11 |
| Logical OR | || |  | 12 |
| a ? x : y means "if a then x, else y” | ? : | Conditional | 13 |
| Simple assignment | = | Assignment | 14 |
| Assign product | \*= |
| Assign quotient | /= |
| Assign remainder (modulus( | =% |
| Assign sum | += |
| Assign difference | -= |