

Infix-Prefix-Postfix

Introduction Part-1

Evaluation of expression

- Expression ဆိုတာ operator တွေ constants တွေ variables များနဲ့ပေါင်းစပ်ဖွဲ့စည်းထားခြင်းဖြစ်ပါတယ်။ Expression ထဲမှာဆို operands တွေ operators တွေပါဝင်ပြီး expression တစ်ခုက value တစ်ခုအမြဲပြန်ပေးပါတယ်။

Eg. $2 + 3 = 5$

- အပေါ်က example မှာ ဆို 2 နဲ့ 3 ရယ်ဆိုပြီး operands 2 ခုပါဝင်ပြီး + operator တစ်ခုပါဝင်တဲ့ expression မှာဆို 5 ဆိုတဲ့ value ကို ရမှာပဲဖြစ်ပါတယ်။

- $2 + 3$
<Operand> <Operator> <Operand>

- အပေါ်က expression မှာဆို မြင်နေကြ structure ဖြစ်တဲ့ operands 2 ခု ရယ် operator တစ်ခုရယ်ပုံစံပဲဖြစ်ပါတယ်။ $2 + 3$ ဆိုတဲ့ expression ကနေရလာတဲ့ value ကိုသိမ်းဖို့ variable တွေသတ်မှတ်ပေးရပါတယ်။

Eg. ကျွန်တော်တို့ sum ဆိုတဲ့ variable တစ်ခုသတ်မှတ်လိုက်ပါမယ်။

$sum = 2 + 3$

$sum = 5$ // ဒါဆို $2+3$ expression ကနေရလာတဲ့ value ကို sum ထဲမှာ assign လုပ်ပြီး သိမ်းထားလိုက်တာပဲဖြစ်ပါတယ်။

Infix

- Expression ပုံစံမျိုးစုံထဲက ခုနမြင်ခဲ့တဲ့ $2 + 3$ ဆိုတဲ့ operand 2 ခုကြား operator တစ်ခုပါဝင်တဲ့ expression မျိုးကို infix notation လို့ခေါ်ပါတယ်။

1. $2+3$

2. $(2 + 3) * 4$

3. $(a + b) * (c + d)$

- Operand ကအမြဲတမ်း constant တွေ variable တွေပဲဖြစ်နိုင်တာမဟုတ်ပါဘူး expression တစ်ခုအနေနဲ့လည်းဖြစ်ရနိုင်ပါတယ်။ နံပါတ် 2 က expression မှာဆို operand 2 ခုကို multiplication လုပ်ထားတာပဲဖြစ်ပါတယ်။ $(2+3)$ ဆိုတဲ့ operand တစ်ခုရယ် 4 ဆိုတဲ့ operand တို့ကြား multiplication operator တစ်ခုပါဝင်မှာဖြစ်ပါတယ်။

Operator precedence rules

- Operator တစ်ခုတည်းရှိတဲ့ expression တစ်ခုကို ဖြေရှင်းဖို့တော့ခက်မှာမဟုတ်ပါဘူး။ $2 + 3$ ဆိုတဲ့ expression တစ်ခုမှာဆို 2 နဲ့ 3 ကိုပေါင်းထားတဲ့ operator တစ်ခုပဲရှိတာကြောင့်လွယ်လွယ်ကူကူဖြေရှင်းလိုရပါတယ်။
- Operator တစ်ခုထက်ပိုတဲ့ expression ($2 + 3 * 4$) မျိုးမှာ ဆို 2 နဲ့ 3 ကိုပေါင်းပြီး 4 နဲ့မြှောက် မလား ၊ 3 နဲ့ 4 ကို မြှောက်ပြီးမှ 2 နဲ့ပေါင်းမလားဆိုပြီး ဘယ် operator ကို အရင်ဆုံးလုပ်ဆောင်ရမလဲဆိုတာကို ကျွန်တော်တို့ ဆုံးဖြတ်ရမှာပဲဖြစ်ပါတယ်။
- ဒီအခြေအနေမျိုးအတွက်ကျွန်တော်တို့ operator precedence rules နဲ့ဖြေရှင်းလိုရပါတယ်။

- Order of operation

1. Parentheses () { } []
2. Exponents // right to left
3. Multiplication and division // left to right
4. Addition and Subtraction. // left to right

Order of operation

- $2 + 3 * 4$ မှာ ဆို operator precedence အရ Multiplication က precedence ပိုမြင်တာကြောင့် 3 နဲ့ 4 ကိုအရင် မြှောက် ပြီး ရလာတဲ့ 12 ကို 2 နဲ့ပေါင်းမှာပဲဖြစ်ပါတယ်။ ဒါဆို operator တစ်ခုထက်ပိုတဲ့အခါ ဘယ် operator ကိုအရင်ဆုံးလုပ်ဆောင်ရမလဲဆိုတာကို operator precedence အရ သိနိုင်မှာဖြစ်ပါတယ်။
- Addition နဲ့ Subtraction ဟာတူညီတဲ့ precedence မှာရှိတာကြောင့် (+ , -) 2ခုလုံး expression တစ်ခုထဲမှာပါ လာမယ်ဆို left to right ပုံစံနဲ့အလုပ်လုပ်သွားမှာဖြစ်ပါတယ်။ ဒါကတော့ expressionတွေမှာတူညီတဲ့ Precedence operator တွေပါလာမယ်ဆို Associativity rule နဲ့သတ်မှတ်ပေးသွားတာပဲဖြစ်ပါတယ်။
- Eg ($2 + 5 - 4$)မှာဆို left to right ပုံစံအရ 2 နဲ့ 5 ကိုအရင်ပေါင်းပြီးရလာတဲ့ 7 ထဲက 4ကိုနုတ်မှာပဲဖြစ်ပါတယ်။
- Infix မှာ ပါတဲ့ expression တစ်ခုကို evaluate လုပ်တဲ့အခါမှာ precedence ကိုအရင်ဆုံး ကြည့်ရမှာပဲဖြစ်ပါတယ်။

အခုလိုမျိုး tips လေးတွေကို ကျွန်တော်တို့ facebook page ဖြစ်တဲ့ code craft မှနေ့စဉ်တင်ပေးမှာဖြစ်တာကြောင့် like and follow လေးတွေလုပ်ထားပေးကြပါခဗျာ။

Source from (Mycodeschool)

အားလုံးကို ကျေးဇူးတင်ပါတယ်ခဗျာ။ <Code craft>