

Bison Ayrıştırıcı Üretici

Windows işletim sistemi kullananlar için alternatifler:

- Windows subsystem Ubuntu
- Cygwin
- Linux on virtual machine

Deney için kurulması gerekli paketler:

- gcc
- bison

Bison'un pcalc.y dosyası ile çalıştırılması adımları:

- bison pcalc.y (Bu komut ile bison "pcalc.tab.c" adlı bir dosya oluştur.)
- gcc -o pcalc pcalc.tab.c -lm (Bir önceki adımda oluşturulan c kodu dosyası derlenir.)
- ./pcalc (Derleme sonucu oluşan program çalıştırılır.)

Deyin Uygulaması ve Rapor:

- Deney föyündeki 10. (Deney Tasarımı ve Uygulaması) ve 11. (Deney Soruları 1 ve 2) başlıklardaki maddeleri uygulayın
- Yylex fonksiyonunu boşluk ve tab karakterlerini atladığı gibi yorum satırlarını da atlmasını sağlayacak şekilde düzenleyin.
 - o Örnek 1: 4 + 5 /* Yorum */
 - o Örnek 2: /* Yorum */ 5 - 4
- Ekran çıktılarını da kullanarak yaptığınız işlemleri ve deneyin kazanımlarını detaylı bir şekilde yazarak hazırlayacağınız raporu **Pazar (26 Nisan) 23:59'a** kadar e-posta atmalısınız.

Not: Deney Tasarımı ve Uygulaması başlığında 9. Madde için aşağıdaki sonuçlar üretilmelidir.

$$4 + 4.5 - (34/(8*3+3)) = 6.880952381$$

$$10 ^ 5 * 8 / 5 + 3 = 160003$$

Email: mehmetcemil@gmail.com

Kaynaklar:

Bison: http://www.gnu.org/software/bison/manual/html_node/index.html

Subsystem Linux: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>

```

%{
#define YYSTYPE double
#include <stdio.h>
#include <math.h>
#include <ctype.h>

int yylex (void);
void yyerror (char const *);
%}

%token NUM

%% /* Grammar rules and actions follow. */
input:
    /* empty */
    | input line
    ;
line:
    '\n'
    | exp '\n' { printf ("\t%.10g\n", $1); }
    ;
exp: NUM { $$ = $1; }
| exp exp '+' { $$ = $1 + $2; }
| exp exp '-' { $$ = $1 - $2; }
| exp exp '*' { $$ = $1 * $2; }
| exp exp '/' { $$ = $1 / $2; }
| exp exp '^' { $$ = pow ($1, $2); } /* Exponentiation */
| exp 'n' { $$ = -$1; } /* Unary minus */

;
%%
/* The lexical analyzer returns a double floating point number on the
stack and the token NUM, or the numeric code of the character read if not
a number. It skips all blanks and tabs, and returns 0 for end-of-input.
*/
int yylex (void){
int c;
/* Skip white space. */
while ((c = getchar ()) == ' ' || c == '\t');
/* Process numbers. */
if (c == '.' || isdigit (c)){
    ungetc (c, stdin);
    scanf ("%lf", &yyval);
    return NUM;
}
/* Return end-of-input. */
if (c == EOF)
    return 0;
/* Return a single char. */
return c;
}
/* Called by yyparse on error. */
void yyerror (char const *s){
    fprintf (stderr, "%s\n", s);
}
int main (void){
    return yyparse ();
}

```

Tablo 1 Pcalc.y - Reverse polish notation calculator.