Lengkapilah parser yang telah anda buat dengan code generator untuk pemanggilan prosedur (procedure call), sehingga dapat mengompilasi program dan menghasilkan kode dalam instruksi mesin objek ST4469 !

Source code	Object code

`Program ex111;`
`Var a,b,c,d;`
 	
`Procedure abcd(i,j,k,l)`
…
 	
`Begin`
…
`End.`
	code untuk deklarasi variabel
	`jmp 0 0 x`
	code untuk procedure
	…
	
	`x:` code untuk main body

Alokasi variable lokal vs variabel global

variabel yang bisa dianggap variabel lokal ada 2 :

- variabel yang dideklarasikan pada prosedur
  area yang dialokasikan untuk variabel ini ada diatas base pointer (register R1) dari stack memory. Untuk memudahkan code generator, setiap pendeklarasian variabel, langsung dilakukan pencatatan alamat relatif area (terhadap base pointer) yang dialokasikan pada variabel tersebut kedalam tabel symbol

  Contoh :

```
procedure xyz(a,b,c,d)
var i,j,k;
```

  untuk variabel `i`, kita alokasikan alamat 0 relatif terhadap R1
  untuk variabel `j`, kita alokasikan alamat 1 relatif terhadap R1
  untuk variabel `k`, kita alokasikan alamat 2 relatif terhadap R1.

- variabel yang merupakan passing parameter dari suatu prosedur atau fungsi
  area yang dialokasikan untuk variabel ini ada dibawah base pointer dari stack memory. Untuk memudahkan code generator, setiap pendeklarasian prosedur terutama pada bagian passing parameter, langsung dilakukan pencatatan alamat relatif area (terhadap base pointer) yang dialokasikan pada passing parameter tersebut kedalam tabel symbol.

  Contoh :

```
procedure xyz(a,b,c,d)
```

  untuk passing parameter d, kita alokasikan alamat -5 relatif terhadap R1
  untuk passing parameter c, kita alokasikan alamat -6 relatif terhadap R1
  untuk passing parameter b, kita alokasikan alamat -7 relatif terhadap R1
  untuk passing parameter a, kita alokasikan alamat -8 relatif terhadap R1

Perlu diperhatikan bahwa nilai relatif dimulai dari -5, karena area stack memory dengan alamat relatif terhadap R1 antara -4 hingga -1 digunakan internal mesin objek untuk menyimpan data-data dari stack frame sebelumnya (untuk lebih jelasnya, perhatikan spesifikasi instruksi **`call`** mesin objek ST4469).

Secara umum, dalam penggunaan variabel lokal, operand *b* selalu bernilai 1, dan nilai *a* adalah alamat relatif terhadap R1 (ingat format instruksi ST4469). Sedangkan untuk variabel global, operand *b* selalu bernilai 0, dan nilai *a* adalah alamat relatif terhadap alamat 0 dari stack memory itu sendiri.

Code generator untuk bagian prosedur

Selain dengan hal yang terkait penggunaan variabel / passing parameter, proses sama dengan code generator sebelumnya.

Code generator untuk bagian pemanggilan prosedur

Source code	Object code

`xyz(`*expression1 expression2, expression3,…*`)`	…
	code untuk *expression1*
	code untuk *expression2*
	code untuk *expression3*
	`cal  0 0 address-xyz`
	`int  0 0 –(jumlah parameter)`

Testing Code :

Input Source	Object Code

program example131;
var n,x;
procedure prime;
var m;
begin
  m:=x div 2;
  while x <> (x div m) *m do
  m:=m-1;
  if m=1 then write(x)
end;
begin
  read(n);
  while 1<n do begin
  x:=n;
  prime;
  n:=n-1;
  end
end.
	int 0 0 2
	jmp 0 0 27
	int 0 0 1
	lod 0 0 1
	lit 0 0 2
	opr 0 0 4
	sto 1 0 0
	lod 0 0 1
	lod 0 0 1
	lod 1 0 0
	opr 0 0 4
	lod 1 0 0
	opr 0 0 3
	opr 0 0 6
	jpc 0 0 20
	lod 1 0 0
	lit 0 0 1
	opr 0 0 2
	sto 1 0 0
	jmp 0 0 7
	lod 1 0 0
	lit 0 0 1
	opr 0 0 5
	jpc 0 0 26
	lod 0 0 1
	put 0 0 0
	rtn 0 0 0
	get 0 0 0
	sto 0 0 0
	lit 0 0 1
	lod 0 0 0
	opr 0 0 7
	jpc 0 0 41
	lod 0 0 0
	sto 0 0 1
	cal 0 0 2
	lod 0 0 0
	lit 0 0 1
	opr 0 0 2
	sto 0 0 0
	jmp 0 0 29
	hlt 0 0 0

```
program example132;

var n,temp;
procedure fact(n);
begin if n<=1 then temp:=1
      else begin
        fact(n-1);
       temp := temp*n
      end
end;

begin
  read(n);
  fact(n);
  write(temp)
end.
```

```
int  0  0   2
jmp  0  0  19
lod  1  0  -5
lit  0  0   1
opr  0  0   8
jpc  0  0   9
lit  0  0   1
sto  0  0   1
jmp  0  0  18
lod  1  0  -5
lit  0  0   1
opr  0  0   2
cal  0  0   2
int  0  0  -1
lod  0  0   1
lod  1  0  -5
opr  0  0   3
sto  0  0   1
rtn  0  0   0
get  0  0   0
sto  0  0   0
lod  0  0   0
cal  0  0   2
int  0  0  -1
lod  0  0   1
put  0  0   0
hlt  0  0   0
```