

Derleyici Tasarımı Kısa Sınav 1

1. Aşağıdaki program parçası için derleyici çıkışı nedir, yorumlayıcı çıkışı nedir?

for (i=1; i<=4; i++) cout << i*3;	
Compiler	Interpreter
<pre> LOD R1,='4' STO R1,Temp1 MOV i,='1' L1: CMP i,Temp1 BH L2 {Branch if i>Temp1} LOD R1,i MUL R1,='3' STO R1,Temp2 PUSH Temp2 CALL Write ADD i,='1' {Add 1 to i} B L1 L2: </pre>	<pre> 3 6 9 12 </pre>

Soru 2. Aşağıdaki c/C++ deyimine ilişkin atomları yazınız.

<pre>while (A<=B) A = A + 1;</pre>	<pre> (LBL, L1) (TEST, A, <=, B, L2) (JMP, L3) (LBL, L2) (ADD, A, 1, A) (JMP, L1) (LBL, L3) </pre>
---------------------------------------	---

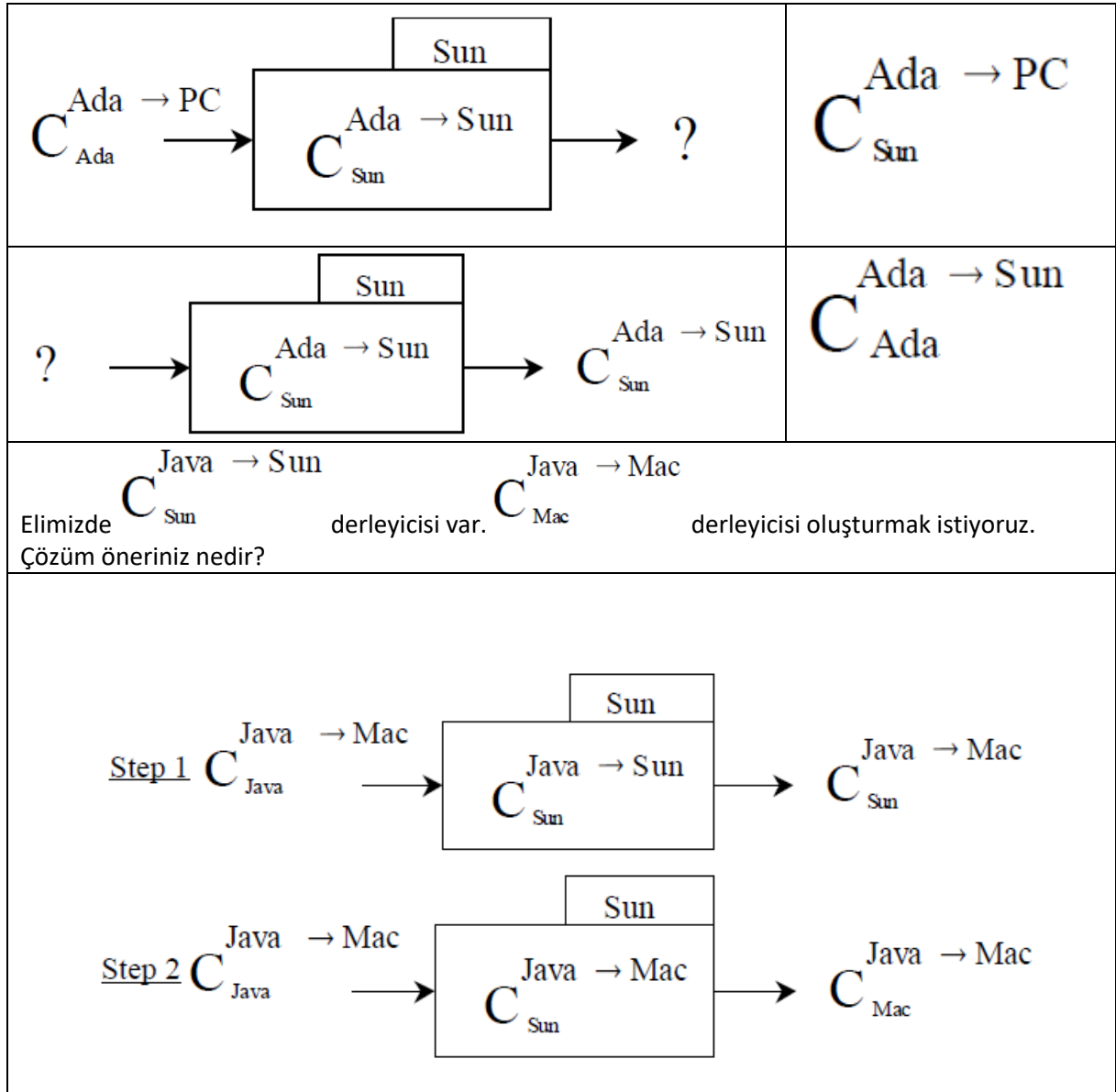
Soru 6. Belirsiz Gramer nedir? Aşağıda verilen gramerde belirsizlik varsa bunu yok ederek belirsizlik içermeyen eşdeğer bir gramer tasarlayınız.

<pre> 1. Expr → Expr + Expr 2. Expr → Expr * Expr 3. Expr → (Expr) </pre>	<pre> 1. Expr → Expr + Term 2. Expr → Term 3. Term → Term * Factor 4. Term → Factor 5. Factor → (Expr) 6. Factor → var 7. Factor → const </pre>
---	---

Soru 3. Aşağıdaki ataom katarına ilişkin assembly komutlarını yazınız.

(ADD, A, B, Temp1)	LOD	R1, A
(TEST, A, ==, B, L1)	ADD	R1, B
(MOVE, Temp1, A)	STO	R1, Temp1
(LBL, L1)	CMP	A, B
(MOVE, Temp1, B)	BE	L1
	MOV	A, Temp1
	L1: MOV	B, Temp1

Soru 4. Aşağıdaki derleme süreçlerini tamamlayınız.



Soru 6. Tam sayı ve kayan noktalı sayı için bir lex programı yazınız.

```
INT      [0-9] +
EXP      ( [eE] [+ -] ? {INT} )
%{

int i;
float f;
%}
%%
{INT}      {sscanf (yytext, "%d", &i);
            cout << "Int" << endl;}
{INT}\\. {INT}? {EXP}? {sscanf (yytext, "%lf", &f);
            cout << "Float" << endl;}
.          ; /* sıfırla */
%%
main ()
{ yylex(); }
yywrap ()
{ }
```

Soru 7. a) Aşağıda verilen gramer için bir pushdown tanıyıcı tasarlayınız. b) Recursive Descent Parser oluşturunuz.

1. $S \rightarrow aSB$ 2. $S \rightarrow b$ 3. $B \rightarrow a$ 4. $B \rightarrow bBa$		a	b	ϵ	<div style="border: 1px solid black; padding: 5px; display: inline-block;">S ▽</div> Initial Stack
	S	Rep (Bsa) Retain	Rep (b) Retain	Reject	
	B	Rep (a) Retain	Rep (aBb) Retain	Reject	
	a	pop Advance	Reject	Reject	
	b	Reject	pop Advance	Reject	
	▽	Reject	Reject	Accept	
	a)				

```

char inp;
void parse ()
{ cin >> inp;
  S ();
  if (inp=='\epsilon') accept();
  else reject();
}

void S ()
{ if (inp=='a') // apply rule 1
  { cin >> inp;
    S ();
    B ();
  } // end rule 1
  else if (inp=='b') cin >> inp; // apply rule 2
  else reject();
}

void B ()
{ if (inp=='a') cin >> inp; // apply rule 3
  else if (inp=='b') // apply rule 4
  { cin >> inp;
    B ();
    if (inp=='a') cin >> inp;
    else reject();
  } // end rule 4
  else reject();
}

```

b)