

## Fordítóprogramok 3. zh

1. Készíts kódsablont az alábbi kifejezések kiértékeléséhez.

a)  $V++$

$V$  egy **32** bites változó. A kifejezés értéke  $V$  **eredeti** értéke, mellékhatásként pedig megnöveli  $V$  értékét 1-gyel. Az eredmény  $eax$ -be töltődjön be.

b)  $++V$

$V$  egy **16** bites változó. A kifejezés megnöveli  $V$  értékét 1-gyel, értéke  $V$  **megnövelés utáni** értéke. Az eredmény  $ax$ -be töltődjön be.

c)  $(int)E$

$E$  egy **8** bites **előjeles** kifejezés. Kerüljön  $eax$ -be  $E$  **32** bitre kiterjesztett értéke.

2. Add meg a Pascal hátultesztelő ciklusának kódsablonját. A ciklusfeltétel az utasítás végrehajtása után értékelődik ki. A ciklus futása akkor szakad meg, amikor a feltétel értéke először lesz igaz.

**repeat**

*<utasítás>*

**until** (*<feltétel>*)

3. Add meg az alábbi C++-kódrészlethez generált kódot. A kódot szigorúan a mellékelt lapon szereplő sablonok alapján kell előállítani.

```
// Globális változók.  
int a = 1, b = 1, i = 1;  
int c; // Nincs kezdőérték.
```

```
// ...
```

```
while (i <= 10)  
{  
    c = a + b;  
    a = b;  
    b = c;  
    i = i + 1;  
}
```

4. Optimalizáld a következő kifejezések kiértékelését:

```
a=b+3; c=b; c=(c+3)*(c+3); a=a*a; c=c*c;
```