System-oriented Programming
Spring 2018

S03

Professor : Philippe Cudré-Mauroux
Assistant : Michael Luggen

Submitted by Sylvain Julmy

Note : the complete source file are available inside the zipped file.

Exercice 1

a)

```
for (low = 0; low <= high; low++)
        printf("%i\n", low);
```

b)

```
if (low > high) goto end;
do
{
    printf("%i\n", low);
    low++;
} while (low <= high);

end:
```

We use a *goto* and an *if* in order to don't go inside the loop if $low > high$.

Exercice 2

a)

```
i = 0;
start:
printf("%i\n",i++);
if (i < n) goto start;
```

b)

```
if (i == 1) goto case1;
if (i == 2) goto case2;
goto caseDefault;

case1 :
printf("case 1 \n");
goto end; // break

case2 :
printf("case 2 \n");

caseDefault:
printf("default case \n");
goto end; // unnecessary, but we follow the code example

end:;
```

c)

```
for (i = 0; i < n; i++)
{
    printf("action 1, i=%i\n", i);
    if (i > 0) goto end;
    printf("action 2, i=%i\n", i);
}
end:;
```

d)

```
for (i = 0; i < n; i++)
{
    printf("action 1, i=%i\n", i);
    if (i > 0) goto loopEnd;
    printf("action 2, i=%i\n", i);
    loopEnd:;
}
```

Note : we have to put a semi-colon after the label declaration if there is no following instructions, ; alone acts for the *nop* instruction.

Exercice 3

Figure 1 show the list of command to execute, as well as some comment, in order to detect the causes of the crash.

```
// run first time to show the error
run
// set breakpoint and add display for both expr i and N
break 16
display i
display N
// rerun and answer yes when asking for reruning the program
run
// stop each time on the breakpoint
c     c     c
// now we clearly see that N/i is 3/0 and would lead to a arithmetic error
c
// and now the following message is display :
Program received signal SIGFPE, Arithmetic exception.
0x00005555555546d0 in main () at ex3/div_zero.c:12
```

Figure 1: List of gdb command to execute in order to clearly show the error from program 1.

```c
1  #include <stdio.h>
2
3  int N = 3;
4
5  int main()
6  {
7     int ctr, i;
8     int res;
9
10    i = N;
11    res = N;
12
13    printf("res  N  i\n");
14    for (ctr = 0; ctr <= N; ++ctr, --i)
15    { // 'ctr <= N' for exercice 5
16       res = N / i;
17       printf("%3i%3i%3i\n", res, N, i);
18    }
19
20    return 0;
21 }
```

Listing 1: C program that would have an arithmetic error, a division by zero.

Exercice 4

Figure 2 show the list of command to execute, as well as some comment, in order to detect the causes of the crash.

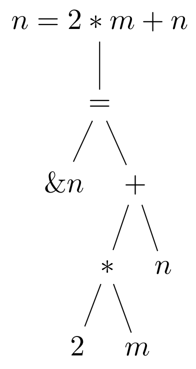
```
// First we create a core file using the generate-core-file command from gdb
// then we load it with (filename of the core file is core.17331)
gdb div_zero core.17331 --tui

// then we can directly watch the varible and saw why the program crashed
display N
display i

// we saw N = 3 and i = 0, then N/i is an arithmetic error :

Core was generated by `/home/snipy/Master/mcs-git/sys-oriented-prog
                       /exercices/s03/exercices/div_zero'.
Program terminated with signal SIGFPE, Arithmetic exception.
#0  0x00005555555546d0 in main () at ex3/div_zero.c:16
(gdb) display N
1: N = 3
(gdb) display i
2: i = 0
(gdb)
```

Figure 2: List of gdb command to execute in order to clearly show the error from program 1 with the line 16 modified.

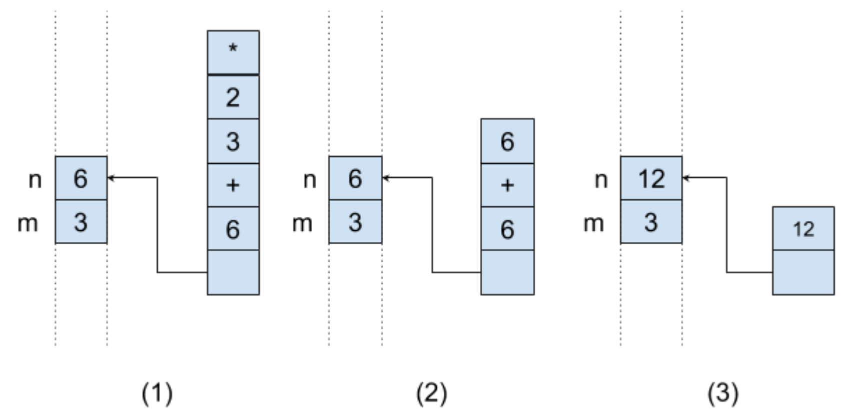
Exercice 5

```
n=2*m+n;
```

Abstract Syntax Tree :

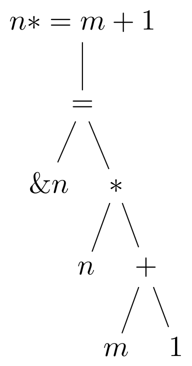


Control Stack :

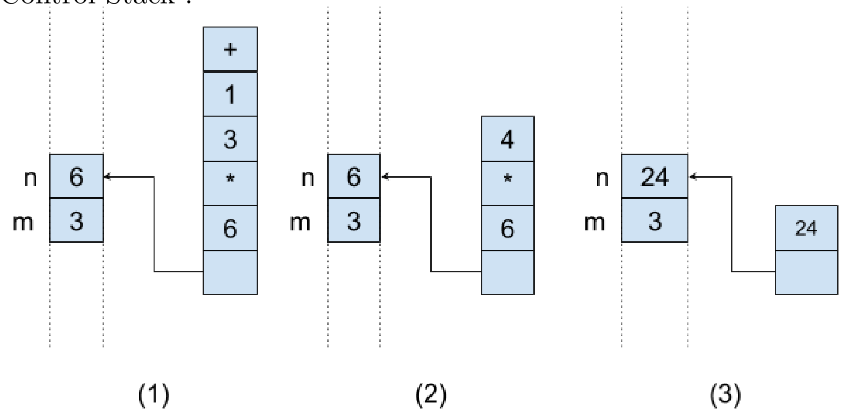


```
n*=m+1
```

Abstract Syntax Tree :



Control Stack :


n
m
6
3
+
1
3
*
6
(1)
n
m
6
3
4
*
6
(2)
n
m
24
3
24
(3)