Buffer Overflow, 1ère partie

1. On considère le programme suivant (example3.c) qui affiche l'adresse de retour et la valeur de x :

```
#include <stdio.h>

void myfunc(int a, int b, int c) {
   char buffer1[2];
   char *ret;

   ret = buffer1 + 14;
   /* (*ret) += ... ; */
   printf("%04x\n",ret);
}

int main(void) {
  int x;

  x = 0;
  myfunc(1,2,3);
  x = 1;
  printf("%d\n",x);
}
```

2. Compiler ce programme avec le Makefile suivant.

```
all: example3
CC=gcc
CFLAGS=-m32 -Wimplicit-function-declaration -O0 -fno-stack-protector \
  -z execstack -mpreferred-stack-boundary=2 -D_FORTIFY_SOURCE=0 \
  -static -ggdb
```

3. À l'aide de GDB, déterminer l'adresse de l'instruction situé à la dernière ligne de la fonction *main*, `printf("%d\n",x)`. Quelle est la distance avec l'adresse de retour qui a été empilée ?
4. Décommenter et compléter la fonction *myfunc* pour que le programme affiche "0" comme valeur pour x. (autrement dit, l'exécution du programme doit être modifiée pour éviter l'affectation "x=1" à l'avant-dernière ligne de la fonction *main*).