FND9

Source code of question 1:

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv)
{
    printf("\nargc = %d\targc address: %p\n", argc, &argc);
    int i;
    for(i=0; i < argc ;i++)
    {
        printf("argument %d: %s\t address: %p\n", i, argv[i], &argv[i]);
    }
    printf("Address of argv: %p\n", &argv);
    printf("Address of *argv: %p\n", &*argv);
    printf("Address of **argv: %p\n", &**argv);
    printf("\n");
    return EXIT_SUCCESS;
}
```

Output

```
[mmoustaf@gc112m38 FND9]$ gcc -o fnd9 fnd9.c
[mmoustaf@gc112m38 FND9]$ ./fnd9 Hello Professor

argc = 3        argc address: 0x7ffcffcf897c
argument 0: ./fnd9      address: 0x7ffcffcf8a78
argument 1: Hello       address: 0x7ffcffcf8a80
argument 2: Professor   address: 0x7ffcffcf8a88
Address of argv: 0x7ffcffcf8970
Address of *argv: 0x7ffcffcf8a78
Address of **argv: 0x7ffcffcfa2b4
```

Question 2

The purpose of "$@" is to evaluate to the current target

The purpose of "$(FLAGS)" is to compile the files with special cases (flags)

New Makefile

```
GCC = gcc
CFLAGS = -g -Wall -Wshadow
OBJS = mystring.o main.o
HDRS = mystring.h
VAL = valgrind --tool=memcheck --leak-check=full
VAL += --verbose --log-file=

mystring: $(OBJS) $(HDRS)
  $(GCC) $(CFLAGS) $(OBJS) -o $@

%o: %.c
  $(GCC) $(CFLAGS) -c $*.c

clean:
  rm -f mystring $(OBJS)
```

New Main

```c
// main.c
#include "mystring.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LINE_SIZE 1000 // a line has at most 999 characters
int my_strlen(const char * str);
int my_countchar(const char * str, char c);
void my_strupper(char * str);
char * my_strchr(const char * str, char ch);
int main(int argc, char *argv[]){
    if (argc != 4) {
        printf("usage: %s command input output\n", argv[0]);
        return EXIT_FAILURE;
    }

    FILE *infptr = fopen(argv[2], "r");
    if (infptr == NULL) {
        printf("unable to open file %s!\n", argv[2]);
        return EXIT_FAILURE;
    }
    FILE *outfptr = fopen(argv[3], "w");
    if (outfptr == NULL) {
        printf("unable to open file %s!\n", argv[3]);
        fclose(infptr);
        return EXIT_FAILURE;
    }
    int num_lines = 0;
    char buffer[LINE_SIZE];

    // count the number of lines in the file
    while (fgets(buffer, LINE_SIZE, infptr) != NULL)
        num_lines++;
```

```
34
35      // return to the beginning of the file
36      fseek(infptr, 0, SEEK_SET);
37
38      char **lines = malloc(sizeof(char *) * num_lines);
39      int i;
40      for (i = 0; i < num_lines; i++) {
41          if (feof(infptr)){
42              printf("not enough num_lines in file!\n");
43              fclose(infptr);
44              fclose(outfptr);
45              return EXIT_FAILURE;
46          }
47          lines[i] = malloc(sizeof(char) * LINE_SIZE);
48          fgets(lines[i], LINE_SIZE, infptr);
49      }
50      fclose(infptr);
51      int total_length = 0;
52      for (i = 0; i < num_lines; i++)
53          total_length += my_strlen(lines[i]);
54
55      // count the length of each line
56      if (strcmp(argv[1], "strlen") == 0) {
57          for (i = 0; i < num_lines; i++){
58              fprintf(outfptr, "length: %d\n",
59              my_strlen(lines[i]));
60          }
61      }
62      /* for each line, count the occurrence of the first
63      letter in the line */
64      if (strcmp(argv[1], "countchar") == 0) {
```

```
64      if (strcmp(argv[1], "countchar") == 0) {
65          for (i = 0; i < num_lines; i++){
66              fprintf(outfptr, "count(%c): %d\n", lines[i][0],
67              my_countchar(lines[i], lines[i][0]));
68          }
69      }
70      if (strcmp(argv[1], "strupper") == 0) {
71          for (i = 0; i < num_lines; i++) {
72              my_strupper(lines[i]);
73              fprintf(outfptr, "%s", lines[i]);
74          }
75      }
76      for (i = 0; i < num_lines; i++)
77          free(lines[i]);
78      free(lines);
79      fclose(outfptr);
80      return EXIT_SUCCESS;
81  }
82
```

Make Output

```
[mmoustaf@gc112m38 FND9]$ make
cc -g -Wall -Wshadow   -c -o main.o main.c
gcc -g -Wall -Wshadow mystring.o main.o -o mystring
[mmoustaf@gc112m38 FND9]$
```