/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DAY 19: COMMAND-LINE PROGRAMS

\* Each program maintains its own includes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 1. Display executable file name \*/

#include <stdio.h>

void program1(int argc, char \*argv[]) {

printf("1. Executable name: %s\n", argv[0]);

}

/\* 2. Print command-line arguments \*/

#include <stdio.h>

void program2(int argc, char \*argv[]) {

printf("2. Arguments (%d):\n", argc-1);

for(int i = 1; i < argc; i++) {

printf(" Arg %d: %s\n", i, argv[i]);

}

}

/\* 3. Sum of command-line numbers \*/

#include <stdio.h>

#include <stdlib.h>

void program3(int argc, char \*argv[]) {

double sum = 0;

for(int i = 1; i < argc; i++) {

sum += atof(argv[i]);

}

printf("3. Sum of arguments: %.2f\n", sum);

}

/\* 4. Check argument count \*/

#include <stdio.h>

void program4(int argc, char \*argv[]) {

printf("4. Number of arguments: %d\n", argc-1);

}

/\* 5. Factorial from command line \*/

#include <stdio.h>

#include <stdlib.h>

void program5(int argc, char \*argv[]) {

if(argc < 2) {

printf("5. Error: Please provide a number\n");

return;

}

int n = atoi(argv[1]);

long fact = 1;

for(int i = 1; i <= n; i++) fact \*= i;

printf("5. Factorial of %d: %ld\n", n, fact);

}

/\* 6. Reverse command-line string \*/

#include <stdio.h>

#include <string.h>

void program6(int argc, char \*argv[]) {

if(argc < 2) {

printf("6. Error: Please provide a string\n");

return;

}

printf("6. Reversed string: ");

for(int i = strlen(argv[1])-1; i >= 0; i--) {

putchar(argv[1][i]);

}

printf("\n");

}

/\* 7. Count vowels in argument \*/

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void program7(int argc, char \*argv[]) {

if(argc < 2) {

printf("7. Error: Please provide a string\n");

return;

}

int vowels = 0;

char \*str = argv[1];

for(int i = 0; i < strlen(str); i++) {

char c = tolower(str[i]);

if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

vowels++;

}

}

printf("7. Vowel count: %d\n", vowels);

}

/\* 8. Compare two strings \*/

#include <stdio.h>

#include <string.h>

void program8(int argc, char \*argv[]) {

if(argc < 3) {

printf("8. Error: Please provide two strings\n");

return;

}

int result = strcmp(argv[1], argv[2]);

printf("8. Comparison result: %s\n",

result == 0 ? "Equal" : result < 0 ? "First is smaller" : "First is larger");

}

/\* 9. Print longest argument \*/

#include <stdio.h>

#include <string.h>

void program9(int argc, char \*argv[]) {

if(argc < 2) {

printf("9. Error: No arguments provided\n");

return;

}

int max\_len = 0, max\_index = 0;

for(int i = 1; i < argc; i++) {

if(strlen(argv[i]) > max\_len) {

max\_len = strlen(argv[i]);

max\_index = i;

}

}

printf("9. Longest argument: %s\n", argv[max\_index]);

}

/\* 10. Average of command-line numbers \*/

#include <stdio.h>

#include <stdlib.h>

void program10(int argc, char \*argv[]) {

if(argc < 2) {

printf("10. Error: Please provide numbers\n");

return;

}

double sum = 0;

for(int i = 1; i < argc; i++) {

sum += atof(argv[i]);

}

printf("10. Average: %.2f\n", sum/(argc-1));

}

/\* Main menu driver \*/

int main(int argc, char \*argv[]) {

if(argc < 2) {

printf("Usage: %s [program\_number] [arguments]\n", argv[0]);

printf("Available programs: 1-10\n");

return 1;

}