DAY 19

1. Write a program to display the name of the executable file.

#include <stdio.h>

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

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

return 0;

}

2. Write a program to accept and print multiple command-line arguments.

#include <stdio.h>

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

printf("Arguments passed:\n");

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

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

}

return 0;

}

3. Write a program to find the sum of numbers passed via command line.

#include <stdio.h>

#include <stdlib.h>

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

int sum = 0;

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

sum += atoi(argv[i]);

}

printf("Sum = %d\n", sum);

return 0;

}

4. Write a program to check number of command-line arguments.

#include <stdio.h>

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

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

return 0;

}

5. Write a program to calculate factorial of a number from command line.

#include <stdio.h>

#include <stdlib.h>

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

if (argc != 2) {

printf("Usage: ./a.out <number>\n");

return 1;

}

int n = atoi(argv[1]);

int fact = 1;

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

fact \*= i;

}

printf("Factorial of %d = %d\n", n, fact);

return 0;

}

6. Write a program to reverse the string passed as a command-line argument.

#include <stdio.h>

#include <string.h>

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

if (argc != 2) {

printf("Usage: ./a.out <string>\n");

return 1;

}

int len = strlen(argv[1]);

for (int i = len - 1; i >= 0; i--) {

printf("%c", argv[1][i]);

}

printf("\n");

return 0;

}

7. Write a program to count vowels in a command-line input string.

#include <stdio.h>

#include <ctype.h>

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

int count = 0;

if (argc != 2) {

printf("Usage: ./a.out <string>\n");

return 1;

}

for (int i = 0; argv[1][i]; i++) {

char ch = tolower(argv[1][i]);

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

count++;

}

}

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

return 0;

}

8. Write a program to compare two command-line input strings.

#include <stdio.h>

#include <string.h>

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

if (argc != 3) {

printf("Usage: ./a.out <string1> <string2>\n");

return 1;

}

if (strcmp(argv[1], argv[2]) == 0)

printf("Strings are equal.\n");

else

printf("Strings are not equal.\n");

return 0;

}

9. Write a program to print longest argument.

#include <stdio.h>

#include <string.h>

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

if (argc < 2) {

printf("No arguments passed.\n");

return 1;

}

int maxLen = 0;

char \*longest = argv[1];

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

if (strlen(argv[i]) > maxLen) {

maxLen = strlen(argv[i]);

longest = argv[i];

}

}

printf("Longest argument: %s\n", longest);

return 0;

}

10. Write a program to calculate average of command-line numbers.

#include <stdio.h>

#include <stdlib.h>

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

if (argc < 2) {

printf("Provide numbers as arguments.\n");

return 1;

}

int sum = 0;

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

sum += atoi(argv[i]);

}

double avg = (double)sum / (argc - 1);

printf("Average = %.2lf\n", avg);

return 0;

}