**COMMAND LINE ARGUMENTS**

./app arg1 arg2

Json file – the keys must be unique ( in JAVA)

Compile with -- gcc -Wall -g cla.c

1 /\*

2 Command line arguments

3

4 main(int argc, char \*argv[]x) //Number of arguments and what are the arguments respectively

5

6 \*/

7

8 #include <stdio.h>

9 #include <stdlib.h>

10 #include <string.h>

11

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

13 {

14 printf("\nNumber of args = %d",argc);

15

16 printf("\n\n");

17 return 0;

18 }

O/P

user50@trainux01:~/practice/try/day16$ ./app 1 2

Number of args = 3

1 /\*

2 Command line arguments

3

4 main(int argc, char \*argv[]x) //Number of arguments and what are the arguments respectively

5

6 \*/

7

8 #include <stdio.h>

9 #include <stdlib.h>

10 #include <string.h>

11

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

13 {

14 int i;

15 printf("\nNumber of args = %d",argc);

16

17 for(i=0;i<argc;i++)

18 printf("\nargv[%d] = %s",i,argv[i]);

19

20 printf("\n\n");

21 return 0;

22 }

~

O/P

user50@trainux01:~/practice/try/day16$ gcc -Wall -g cla.c -o app

user50@trainux01:~/practice/try/day16$ ./a.out

Number of args = 1

user50@trainux01:~/practice/try/day16$ ./app 1 2

Number of args = 3

argv[0] = ./app

argv[1] = 1

argv[2] = 2

Check in github for different alterations

WAP to read the contents of employees from the file store them in the employee records respectively using command line arguments and store it in the structure and print