args.c

```c
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main(int argc, char* argv[])
5  {
6      printf("argc = %d\n", argc);
7      return 0;
8  }
```

```
PaulProgramming: ls
args.c
PaulProgramming: gcc args.c -o add
PaulProgramming: ls
add     args.c
PaulProgramming: ./add
argc = 1
PaulProgramming: ./add paul programming
argc = 3
PaulProgramming: ./add paul
argc = 2
PaulProgramming:
```

03:03   10:10

args.c

```c
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main(int argc, char* argv[])
5  {
6      int i;
7      printf("argc = %d\n", argc);
8      printf("Let's see what is in argv[]\n");
9
10     for(i = 0; i < argc; i++)
11     {
12         printf("argv[%d] = %s\n", i, argv[i]);
13     }
14
15     return 0;
16 }
```

```
PaulProgramming: gcc args.c -o add
PaulProgramming: ./add
argc = 1
Let's see what is in argv[]
argv[0] = ./add
PaulProgramming: ./add 4 6 8
argc = 4
Let's see what is in argv[]
argv[0] = ./add
argv[1] = 4
argv[2] = 6
argv[3] = 8
PaulProgramming:
```

06:07   10:10

Media   Playback   Audio   Video   Subtitle   Tools   View   Help

args.c

```c
1   #include <stdlib.h>
2   #include <stdio.h>
3
4   int main(int argc, char* argv[])
5   {
6       int i, sum = 0;
7       printf("argc = %d\n", argc);
8       printf("Let's see what is in argv[]\n");
9       if(argc > 1)
10      {
11          for(i = 1; i < argc; i++)
12          {
13              printf("argv[%d] = %s\n", i, argv[i]);
14              sum += atoi(argv[i]);
15          }
16          printf("Total = %d\n", sum);
17      }
18      return 0;
19  }
```

ARGS — bash — 49×32

```
PaulProgramming: gcc args.c -o add
PaulProgramming: ./add
argc = 1
Let's see what is in argv[]
PaulProgramming: ./add 3 2 5
argc = 4
Let's see what is in argv[]
argv[1] = 3
argv[2] = 2
argv[3] = 5
Total = 10
PaulProgramming:
```

09:16   10:10