Q.4 :→ Explain what is happening in each line of the program.

```
#include <stdio.h>
void main (int argn, char *args[])
// Calling the main function. The arguments are : 1) argn of int type. 2) args of which is an array character pointer.
{ int i; // Declares the variable i of integer data type
  FILE *fp; // Declares the variable fp, which points at the
              FILE data type.

  fp = fopen ("note.txt", "a"); // fopen() is a library function
  that is used to open (or create) a file with different modes
  of access. It requires 2 parameters (arguments) :→
    1. file name :→ type : string ⇒ accepts the name of file.
    2. mode of operation :→ type: string ⇒ "a" refers to
       the mode which opens the file for appending.
  // Here, the file name is "note.txt" and "a" means that
    the file is opened to add data to it.
  if (argn > 2) // Check if the number of arguments given at
                   the command line is greater than two.

  { fprintf (fp, "**** %s **** \n", args[1]);
  // This prints the second argument given by user in the
    command line in place of %s. fprintf function is used
    to print content in the file instead of in the stdout console.
  for (i=2; i<argn; i++)
  {
    if (i%10==0)
      { fprintf (fp, "\n"); // Changes line after every
                               10 arguments.
      }

    fprintf (fp, "%s", args[i]); // Prints the ith argument
    starting from i=2. The first argument (which is program
    file name) is not printed.
  }
```

```
        fprintf (fp, "\n.....\n");
        // prints a sequence of dashes after printing the arguments.
    }
    fclose(fp);
    // The file corresponding to the pointer fp is closed, here
       the name of the file is note.txt.
}
```

Running the program :→

```
gcc Q4.c
```

① ./a.exe I am Sai Kaarthik.

Output: (Inside note.txt)

```
**** I *****
amSaikaarthik.
```

② ./a.exe Sai

Output :→ Blank since the number of argument is not greater than equal to 2.

③ ./a.exe a b c d e f g h i j k l m

Output :→

```
**** a *****
bcdefghi
jklm
```

(Since the codition i % 10 == 0 holds for this case which changes line after every 10 arguments.)
