```c
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    char msg[80];
    int stat;
    char *result2=NULL;
    char *argv2[3];
    if(argc>1 && strcmp(argv[1], "-debug" ) == 0)
    {
        printf("INFO: Father started PID[%d]\n",getppid());
    }
    printf("Welcome in my mini shell. Type exit to terminate.\n");
    printf("minishell>");
    gets(msg);
    while( strcmp( msg, "exit" ) != 0 )
    {

        switch (fork())
        {
          case -1: // error
            perror("fork call");
            exit(1);
          case 0:
            if(argc>1 && strcmp(argv[1], "-debug" ) == 0)
            {
                printf("INFO: Child started PID[%d] command %s\n",getppid(),msg);
            }
            result2 = strtok(msg," ");
                            int i;
                            i=0;
                            while( result2 != NULL )
                            {
                            argv2[i]=result2;
                             i++;
                             result2 = strtok(NULL, "" );
                            }

                             argv2[i]=NULL;
                            execvp(argv2[0], argv2);

            exit(1);
          default:
            wait(stat);
            printf("minishell>");
            gets(msg);
        }

    }
}
```

```c
    if(argc>1 && strcmp(argv[1], "-debug" ) == 0)
            {
        printf("INFO: Father will terminate\n");
            }

    }
```