```
/*compilier: gcc -Wall -o bonus bonus.c bonus.h
Execution:./bonus
Student: Frank Wu
Assignment: Bonus Homework
*/


#include  <stdio.h>
#include  <sys/types.h>
#include "bonus.h"

//necessary headers implemented which include the bonus header
void  parse(char *line, char **argv)
{
     while (*line != ’\0’) {       /* if not the end of line ....... */
          while (*line == ’ ’ || *line == ’\t’ || *line == ’\n’)
               *line++ = ’\0’;     /* replace white spaces with 0    */
          *argv++ = line;          /* save the argument position     */
          while (*line != ’\0’ && *line != ’ ’ &&
                 *line != ’\t’ && *line != ’\n’)
               line++;             /* skip the argument until ...    */
     }
     *argv = ’\0’;                 /* mark the end of argument list  */
}
//Referecne http://www.csl.mtu.edu/cs4411.ck/www/NOTES/process/fork/exec.html
//Read the input line and parase into tokens
// Start from replace space into zero, till hittig a non-white space. address were save
d to argv[]
void  execute(char **argv)
{
     pid_t  pid;
     int    status;

     if ((pid = fork()) < 0) {
          printf("*** ERROR: forking child process failed\n");
          exit(1);
     }
     else if (pid == 0) {
          if (execvp(*argv, argv) < 0) {
               printf("\n");
               exit(1);
          }
     }
     else {
          while (wait(&status) != pid)
               ;
     }
}
//Referecne http://www.csl.mtu.edu/cs4411.ck/www/NOTES/process/fork/exec.html
//Recive  command line agument list,
//Initially, being a file name followed by augments
//forks child process. Excecute command with execvp
#define Buff_Len 1024
void  main(void)
{
     char  line[Buff_Len];
     char  *argv[100];
     char path = "/bin/";
     char progpath[20];

     int i,n;
     int arr[100]={0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,983};
     while (1) {
          printf("some fuctions include\n");
          printf("quit, hello, FIB ,fibonacci\n");
          printf("ls, help, cd, \n");
          printf("uab_sh >> ");
//introduction about function and  excecute uab_sh >
          gets(line);
          printf("\n");
          parse(line, argv);
//The above part are from  the Referecne http://www.csl.mtu.edu/cs4411.ck/www/NOTES/pro
cess/fork/exec.html
// and following are commands
          if (strcmp(argv[0], "quit") == 0)  //Exit from the OS
              { exit(0);}
          else if (strcmp(argv[0], "hello") == 0)
              {printf("Hello_World\n");
              execute(argv);
              }
          else if (strcmp(argv[0], "FIB") == 0)
              {printf("The first 10 values is: ");
              for (i=0;i<10;i++){printf("%d, ",arr[i]);}
              printf("\n");
              execute(argv);
              }
          else if (strcmp(argv[0], "fibonacci") == 0 )
              {printf("Key in the amount of numbers to display :\n");
              scanf("%d",&n);
              printf("The first %d values is: ",n);
              for (i=0;i<n;i++){printf("%d ,",arr[i]);}
              execute(argv);}
          else if (strcmp(argv[0], "ls") == 0)
              {
                execvp(argv[1],argv[0]);
                execute(argv);
              }
          else if (strcmp(argv[0], "help") == 0)
              {execute(argv);}

          else if (strcmp(argv[0], "cd") == 0)
              {path = ".";}





     }
}
// Put functions which include cd, help,ls, quit and fibonacci.
```