```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <strings.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>

#define SERVER_PORT  5432
#define MAX_PENDING  5
#define MAX_LINE     256

int main()
{
  struct sockaddr_in sin;
  char buf[MAX_LINE];
  int buf_len, addr_len;
  int s, new_s;
  int rval;
  int choice;

  srand(time(NULL));

  /* build address data structure */
  bzero((char *)&sin, sizeof(sin));
  sin.sin_family = AF_INET;
  sin.sin_addr.s_addr = INADDR_ANY;
  sin.sin_port = htons(SERVER_PORT);

  /* setup passive open */
  if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
    perror("simplex-talk: socket");
    exit(1);
  }
  if ((bind(s, (struct sockaddr *)&sin, sizeof(sin))) < 0) {
    perror("simplex-talk: bind");
    exit(1);
  }
  listen(s, MAX_PENDING);

  /* wait for connection, then receive and print text */
  const char *replies1[5];
  replies1[0] = "Tell me more...";
  replies1[1] = "I’m sorry to hear that.";
  replies1[2] = "How does that make you feel?";
  replies1[3] = "Is it really?";
  replies1[4] = "Are you sure?";

  const char *replies2[3];
  replies2[0] = "What makes you think";
  replies2[1] = "How long was it until";
  replies2[2] = "Did you come to me because";

  printf("Waiting for client connections...\n");
  int client_number = 1;
  while(1) {
    if ((new_s = accept(s, (struct sockaddr *)&sin, &addr_len)) < 0) {
```

```
      perror("simplex-talk: accept");
      exit(1);
    }
    printf("Client %d connected.\n", client_number);
    while (buf_len = recv(new_s, buf, sizeof(buf), 0)) {
      fputs(buf, stdout);

      /* Decide if adding onto string or choosing pre-generated reply*/
      choice = (rand() % (1 - 0 + 1)) + 0;

      if (choice == 1) {
        rval = (rand() % (4 - 0 + 1)) + 0;
        printf("%s\n", replies1[rval]);
      }

      if (choice == 0) {
        rval = (rand() % (2 - 0 + 1)) + 0;
        printf("%s", replies2[rval]);
        printf(" ");
        fputs(buf, stdout);
      }
    }
    close(new_s);
    client_number++;
  }
}
```