```c
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>

int main(void)
{
    int sockfd = 0;
    int bytesReceived = 0;
    char recvBuff[256];
    unsigned char buff_offset[10];   // buffer to send the File offset value
    unsigned char buff_command[2];   // buffer to send the Complete File (0) or Partial File Command (1).
    int offset;                      // required to get the user input for offset in case of partial file command
    int command;                     // required to get the user input for command
    memset(recvBuff, '0', sizeof(recvBuff));
    struct sockaddr_in serv_addr;

    /* Create a socket first */
    if((sockfd = socket(AF_INET, SOCK_STREAM, 0))< 0)
    {
        printf("\n Error : Could not create socket \n");
        return 1;
    }

    /* Initialize sockaddr_in data structure */
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(5001); // port
    serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    /* Attempt a connection */
    if(connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr))<0)
    {
        printf("\n Error : Connect Failed \n");
        return 1;
    }


    /* Create file where data will be stored */
        FILE *fp;
        fp = fopen("destination_file.txt", "ab");
        if(NULL == fp)
        {
          printf("Error opening file");
            return 1;
        }
        fseek(fp, 0, SEEK_END);
        offset = ftell(fp);
        fclose(fp);
        fp = fopen("destination_file.txt", "ab");
        if(NULL == fp)
        {
          printf("Error opening file");
            return 1;
        }

    printf("Enter (0) to get complete file, (1) to specify offset, (2) calculate the offset value from local file\n");
    scanf("%d", &command);
    sprintf(buff_command, "%d", command);
    write(sockfd, buff_command, 2);
```

```
66
67
68
69      if(command == 1 || command == 2)   // We need to specify the offset
70      {
71
72          if(command == 1)  // get the offset from the user
73          {
74          printf("Enter the value of File offset\n");
75          scanf("%d", &offset);
76          }
77          // otherwise offset = size of local partial file, that we have
    already calculated
78          sprintf(buff_offset, "%d", offset);
79          /* sending the value of file offset */
80          write(sockfd, buff_offset, 10);
81      }
82
83      // Else { command = 0 then no need to send the value of offset }
84
85
86      /* Receive data in chunks of 256 bytes */
87      while((bytesReceived = read(sockfd, recvBuff, 256)) > 0)
88      {
89          printf("Bytes received %d\n",bytesReceived);
90          // recvBuff[n] = 0;
91          fwrite(recvBuff, 1,bytesReceived,fp);
92          // printf("%s \n", recvBuff);
93      }
94
95      if(bytesReceived < 0)
96      {
97          printf("\n Read Error \n");
98      }
99
100
101     return 0;
102 }
```