```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
#define BUFFERSIZE 1024
void main(){
      int client , newsocket;
      char buffer[BUFFERSIZE];
      struct sockaddr_in serv_addr;
      struct sockaddr_storage store;
      socklen_t addr_size;
      client = socket(AF_INET,SOCK_STREAM,0);
      serv_addr.sin_family = AF_INET;
            serv_addr.sin_port = htons(6266);
      serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
      connect(client,(struct sockaddr *)&serv_addr ,sizeof(serv_addr));
      strcpy(buffer , "Hi,Server\n");
      send(client , buffer , 11 , 0);
      recv(client , buffer , BUFFERSIZE , 0);
      printf("[recieved]\t%s\n",buffer);
      close(client);
}
```

```c
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#define BUFFERSIZE 1024
void main(){
      int server , newsocket;
      char buffer[BUFFERSIZE];
      struct sockaddr_in serv_addr;
      struct sockaddr_storage store;
      socklen_t addr_size;
      server = socket(AF_INET,SOCK_STREAM,0);
      serv_addr.sin_family = AF_INET;
            serv_addr.sin_port = htons(6266);
      serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
      bind(server,(struct sockaddr*)&serv_addr,sizeof(serv_addr));
      if(listen(server,5) == 0){
            printf("[info]\tlistening to connections...\n");
      }else{
            printf("[error]\tunable to listen \n");
      }
      newsocket = accept(server , (struct sockaddr*)&store ,&addr_size);
      recv(newsocket , buffer  , BUFFERSIZE,0);
      printf("[recieved]\t%s\n",buffer);
      strcpy(buffer,"Hi,Client\n");
      printf("[info]\tSending data to client\n");
      send(newsocket,buffer,11,0);
      close(newsocket);
      close(server);
      return;
}
```

```
❱ ./server                                    ❱ ./client
[info]      listening to connections...       [recieved]  Hi,Client
[recieved]  Hi,Server

[info]      Sending data to client
```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#define BUFFERSIZE 1024

void main(){
      int server;
      char cli_msg[BUFFERSIZE] , serv_msg[BUFFERSIZE];
      struct sockaddr_in serv_addr,client_addr;
      int client_struct_length = sizeof(serv_addr);
      struct sockaddr_storage store;
      socklen_t addr_size;
      server = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
      serv_addr.sin_family = AF_INET;
      serv_addr.sin_port = htons(6265);
      serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
      strcpy(cli_msg , "Hello");
      int sentstatus = sendto(server,cli_msg,sizeof(cli_msg),0,(struct sockaddr
                  *)&serv_addr,client_struct_length);
      if(sentstatus < 0){
            printf("[debug] %d\n",sentstatus);
            printf("[error] can't send \n");
            exit(-1);
      }else{
            printf("[info] Message sent\n");
            printf("[debug] %d\n",sentstatus);
      }
      int recvstat = recvfrom(server,serv_msg,sizeof(serv_msg),0,(struct
            sockaddr *)&serv_addr,&client_struct_length);
      printf("[debug] %d\n",recvstat);
      if(recvstat < 0){
            printf("[error] can't recieve \n");
            exit(-1);
      }else{
            printf("[recieved] %s\n",serv_msg);
      }
      close(server);
      return;
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#define BUFFERSIZE 1024

void main(){
      int server;
      char cli_msg[BUFFERSIZE] , serv_msg[BUFFERSIZE];
      struct sockaddr_in serv_addr,client_addr;
      int client_struct_length = sizeof(client_addr);
      struct sockaddr_storage store;
      socklen_t addr_size;
      printf("[breakpoint 1]\n");
      server = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
      printf("[breakpoint 2]\n");
      serv_addr.sin_family = AF_INET;
      serv_addr.sin_port = htons(6265);
      serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
      int binded = bind(server,(struct sockaddr*)&serv_addr,sizeof(serv_addr));
      if(binded < 0){
            printf("[err] cannot bind\n");
      }else{
            printf("[info] binded\n");
      }
      if(recvfrom(server,cli_msg,sizeof(cli_msg),0,(struct sockaddr *
            )&client_addr,&client_struct_length)<0){
            printf("[error] can't recieve \n");
            exit(-1);
      }
      printf("[recieved] %s\n",cli_msg);
      strcpy(serv_msg , cli_msg);
      if(sendto(server,serv_msg,sizeof(cli_msg),0,(struct sockaddr
                  *)&client_addr,client_struct_length)<0){
            printf("[error] can't send \n");
            exit(-1);
      }
      printf("[info] Message sent\n");
      close(server);
      return;
}
```

```
❱ ./server                    ❱ ./client
[info] binded                 [info] Message sent
[recieved] Hello              [recieved] Hello
[info] Message sent
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define WINDOW_SIZE 5
#define BUFFER_SIZE 1024
int main(){
	int server , client , k=WINDOW_SIZE, m=0 , p;
	char buffer[BUFFER_SIZE];
	struct sockaddr_in server_addr,store;
	socklen_t addr_size = sizeof(server_addr);
	server = socket(AF_INET,SOCK_STREAM,0);
	server_addr.sin_family = AF_INET;
	server_addr.sin_port = htons(5600);
	server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
	memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
	bind(server , (struct sockaddr*)&server_addr ,sizeof(server_addr));
	if((listen(server,5) == 0))
		printf("[info] listening...\n");
	client = accept(server , (struct sockaddr*)&store , &addr_size);
	if(client != -1)
		printf("[info] client is accepted\n");
	while(k != 0){
		printf("[info] Waiting for frame %d\n",m);
		int recv_stat = -1;
		recv_stat = recv(client , buffer , BUFFER_SIZE , 0 );
		if(strncmp(buffer,"frame" , 5) == 0){
			char atoi_buff[2];
			atoi_buff[0] = buffer[6];
			atoi_buff[1] = '\0';
			if( atoi(atoi_buff) == m ){
				printf("[recv] Frame %c\n",buffer[6]);
				strcpy(buffer, "ACK x");
				buffer[4] = ((char)m+1 + '0');
				sleep(1);
				int sent_stat = -1;
				sent_stat = send(client,buffer,6,0);
				if( sent_stat == -1){
					printf("[error] Cannot send ACK : %s\n",buffer);
					exit(-3);
				}
				m++;
				k--;
			}else{
				printf("[recv] Expected %d but got %s discarding ...\
					n",m,buffer);
			}
		}else{
			printf("[recv] Not a frame \n");
		}
	}
	close(client);
	close(server);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define WINDOW_SIZE 5
#define BUFFER_SIZE 1024
int main(){
	int client , k=WINDOW_SIZE , m = 0;
	char buffer[1024] , ack_number_buffer[2];
	ack_number_buffer[0] = '0';
	ack_number_buffer[1] = '\0';
	struct sockaddr_in server_addr;
	socklen_t addr_size = sizeof(server_addr);
	client = socket(AF_INET , SOCK_STREAM , 0);
	server_addr.sin_family = AF_INET;
	server_addr.sin_port = htons(5600);
	server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
	memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
	int conn_stat = connect(client , (struct sockaddr*)&server_addr , addr_size);
	while(k != 0){
		if(m<=WINDOW_SIZE){
			if(atoi(ack_number_buffer) == m){
				strcpy(buffer , "frame x");
				buffer[6] = ((char)m + '0');
				int send_stat = -1;
				send_stat = send(client,buffer,8,0);
				if(send_stat != -1)
				    printf("[send] %s\n",buffer);
				int recv_stat = -1;
				recv_stat = recv(client,buffer,BUFFER_SIZE,0);
				char atoi_buff[2];
				atoi_buff[0] = buffer[4];
				atoi_buff[1] = '\0';
				printf("[recv] recieved ACK : %s\n",buffer);
				m = atoi(atoi_buff);
				ack_number_buffer[0] = buffer[4];
				k--;
			}else{
				printf("[error] Expected ACK %d but have ack %s" , m ,
ack_number_buffer);
			}
		}
	}
	close(client);
}
```

```
❱ ./server
[info] listening...
[info] client is accepted
[info] Waiting for frame 0
[recv] Frame 0
[info] Waiting for frame 1
[recv] Frame 1
[info] Waiting for frame 2
[recv] Expected 2 but got 1 discarding
[recv] Frame 2
```

```
❱ ./client
[send] frame 0
[recv] recieved ACK : ACK 1
[send] frame 1
[error] Expected ACK 2 but have ack 1
[send] frame 1
[recv] recieved ACK : ACK 2
[send] frame 2
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define WINDOW_SIZE 5
#define BUFFER_SIZE 1024
int main(){
        int client , k=WINDOW_SIZE , m = 0;
        char buffer[1024] , ack_number_buffer[2];
        ack_number_buffer[0] = '0';
        ack_number_buffer[1] = '\0';
        int sim_err = 0;
        struct sockaddr_in server_addr;
        socklen_t addr_size = sizeof(server_addr);
        client = socket(AF_INET , SOCK_STREAM , 0);
        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5600);
        server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
        memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
        int conn_stat = connect(client , (struct sockaddr*)&server_addr , addr_size);
        if(conn_stat != -1)
           printf("Connected to %d\n",conn_stat);
        while(k != 0){
                if(m<=WINDOW_SIZE){
                        if(sim_err == 1 && m == 2){
                                sim_err = 0;
                                m++;
                                continue;
                        }
                        if(atoi(ack_number_buffer) == m){
                                strcpy(buffer , "frame x");
                                buffer[6] = ((char)m + '0');

                                int send_stat = -1;
                                send_stat = send(client,buffer,8,0);
                                if(send_stat == -1){
                                        printf("[error] cannot send frame \n");
                                        exit(-2);
                                }else{
                                        printf("[send] %s\n",buffer);
                                }
                                int recv_stat = -1;
                                recv_stat = recv(client,buffer,7,0);
                                if(recv_stat == -1){
                                        printf("[error] cannot recieve any frame \n");
                                        m++;
                                }
                                if(strncmp(buffer,"NACK",4) == 0){
                                        char atoi_buff[2];
                                        atoi_buff[0] = buffer[5];
                                        atoi_buff[1] = '\0';
                                        printf("[recv] recieved NACK : %s\n",buffer);
                                        m = atoi(atoi_buff);
                                        ack_number_buffer[0] = buffer[5];
                                }else{
                                        char atoi_buff[2];
                                        atoi_buff[0] = buffer[4];
                                        atoi_buff[1] = '\0';
                                        printf("[recv] recieved ACK : %s\n",buffer);
                                        m = atoi(atoi_buff);
                                        ack_number_buffer[0] = buffer[4];
                                }
                                k--;
                        }else{
                                printf("[error] Expected ACK %d but have ack %s\n" , m , ack_number_buffer);
                                m = 0;
                                ack_number_buffer[0] = '0';
                                k = WINDOW_SIZE+1;
                        }
                }
        }
        close(client);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/wait.h>
#define BUFFER_SIZE 2048
#define ERR_LISTEN -1
#define ERR_RECV -2
#define ERR_CONN -3
int main(){
        int server,client;
        char recv_buffer[BUFFER_SIZE];
        struct sockaddr_in server_addr,client_addr;
        socklen_t addr_size = sizeof(server_addr);
        server = socket(AF_INET,SOCK_STREAM,0);
        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5600);
        server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
        memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
        bind(server , (struct sockaddr*)&server_addr ,sizeof(server_addr));
        if((listen(server,5) == 0))
                printf("[info] listening... \n");
        client = accept(server , (struct sockaddr*)&client , &addr_size);
        if(client != -1)
           printf("[info] client [%d] is accepted\n",client);
        int buffer[] = {0,1,2,3,4,5};
        const int window_size = 6;
        int recv_buffer_0[window_size];
        int cursor = 0;
        int back = 0;
        int isBufferMessedUp = 0;
        int NACK_req = -1;
        int expected_packet = 0;
        int waiting_bfr[window_size];
        while(expected_packet < window_size){
                printf("[info] waiting for frame %d\n",expected_packet);
                int recv_stat = -1;
                recv_stat = recv(client,recv_buffer,BUFFER_SIZE,0);
                if(recv_stat == -1){
                        printf("[error] cannot recieve packet\n");
                        exit(ERR_RECV);
                }
                if(strncmp(recv_buffer,"frame",5) == 0){
                        char atoi_buff[2];
                        atoi_buff[0] = recv_buffer[6];
                        atoi_buff[1] = '\0';
                        if(expected_packet == atoi(atoi_buff)){
                                expected_packet++;
                                printf("[info] Recieved Packet %s\n",atoi_buff);
                        }else{
                                printf("[info] Got Frame %s instead of %d\n",atoi_buff,expected_packet);
                                waiting_bfr[back] = atoi(atoi_buff);
                                NACK_req =expected_packet;
                        }
                        if(NACK_req != -1){
                                char buff[7];
                                strcpy(buff,"NACK X");
                                buff[5] = recv_buffer[6];
                                send(client,buff,7,0);
                                printf("[info] send %s\n",buff);
                                NACK_req = -1;
                        }else{
                                char buff[] = "ACK X";
                                buff[4] = recv_buffer[6] + 1;
                                send(client,buff,7,0);
                                printf("[info] send %s\n",buff);
                        }
                }
        }

}
```

```
❱ ./server
[info] listening...
[info] client [4] is accepted
[info] waiting for frame 0
[info] Recieved Packet 0
[info] send ACK 1
[info] waiting for frame 1
[info] Recieved Packet 1
[info] send ACK 2
[info] waiting for frame 2
[info] Recieved Packet 2
[info] send ACK 3
[info] waiting for frame 3
[info] Recieved Packet 3
[info] send ACK 4
[info] waiting for frame 4
[info] Recieved Packet 4
[info] send ACK 5
[info] waiting for frame 5
[info] Got Frame 4 instead of 5
[info] send NACK 4
[info] waiting for frame 5
[info] Got Frame 4 instead of 5
```

```
❱ ./client
Connected to 0
[send] frame 0
[recv] recieved ACK : ACK 1
[send] frame 1
[recv] recieved ACK : ACK 2
[send] frame 2
[recv] recieved ACK : ACK 3
[send] frame 3
[recv] recieved ACK : ACK 4
[send] frame 4
[recv] recieved ACK : ACK 5
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define WINDOW_SIZE 5
#define BUFFER_SIZE 1024
int main(){
	int client , k=WINDOW_SIZE , m = 0;
	char buffer[1024] , ack_number_buffer[2];
	ack_number_buffer[0] = '0';
	ack_number_buffer[1] = '\0';
	int sim_err = 1;
	struct sockaddr_in server_addr;
	socklen_t addr_size = sizeof(server_addr);
	client = socket(AF_INET , SOCK_STREAM , 0);
	server_addr.sin_family = AF_INET;
	server_addr.sin_port = htons(5600);
	server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
	memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
	int conn_stat = connect(client , (struct sockaddr*)&server_addr , addr_size);
	while(k != 0){
		if(m<=WINDOW_SIZE){
			if(sim_err == 1 && m == 2){
				sim_err = 0;
				m++;
				continue;
			}
			if(atoi(ack_number_buffer) == m){
				strcpy(buffer , "frame x");
				buffer[6] = ((char)m + '0');

				int send_stat = -1;
				send_stat = send(client,buffer,8,0);
				if(send_stat == -1){
					printf("[error] cannot send frame \n");
					exit(-2);
				}else{
					printf("[send] %s\n",buffer);
				}
				int recv_stat = -1;
				recv_stat = recv(client,buffer,BUFFER_SIZE,0);
				if(recv_stat == -1){
					printf("[error] cannot recieve any frame \n");
					exit(-3);
				}
				char atoi_buff[2];
				atoi_buff[0] = buffer[4];
				atoi_buff[1] = '\0';
				printf("[recv] recieved ACK : %s\n",buffer);
				m = atoi(atoi_buff);
				ack_number_buffer[0] = buffer[4];
				k--;
			}else{
				printf("[error] Expected ACK %d but have ack %s\n" , m ,
						ack_number_buffer);
				m = 0;
				ack_number_buffer[0] = '0';
				k = WINDOW_SIZE+1;
			}
		}
	}
	close(client);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define WINDOW_SIZE 5
#define BUFFER_SIZE 1024
int main(){
	int server , client , k=WINDOW_SIZE, m=0 , p;
	char buffer[BUFFER_SIZE];
	int sim_err = 1;
	struct sockaddr_in server_addr,store;
	socklen_t addr_size = sizeof(server_addr);
	server = socket(AF_INET,SOCK_STREAM,0);
	server_addr.sin_family = AF_INET;
	server_addr.sin_port = htons(5600);
	server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
	memset(server_addr.sin_zero,'\0',sizeof(server_addr.sin_zero));
	bind(server , (struct sockaddr*)&server_addr ,sizeof(server_addr));
	if((listen(server,5) == 0)){
		printf("[info] listening...\n");
	}
	client = accept(server , (struct sockaddr*)&store , &addr_size);
	while(k != 0){
		printf("[info] Waiting for frame %d\n",m);
		int recv_stat = -1;
		recv_stat = recv(client , buffer , BUFFER_SIZE , 0 );
		if(strncmp(buffer,"frame" , 5) == 0){
			char atoi_buff[2];
			atoi_buff[0] = buffer[6];
			atoi_buff[1] = '\0';
			if( atoi(atoi_buff) == m ){
				printf("[recv] Frame %c\n",buffer[6]);
				strcpy(buffer, "ACK x");
				buffer[4] = ((char)m+1 + '0');
				sleep(1);
				int sent_stat = -1;
				sent_stat = send(client,buffer,6,0);
				m++;
				k--;
			}else{
				printf("[recv] Expected %d but got %s discarding ...\n",m,buffer);
				strcpy(buffer, "ACK 0");
				sleep(1);
				int sent_stat = -1;
				sent_stat = send(client,buffer,6,0);
				k = WINDOW_SIZE;
				m = 0;
			}
		}else{
			printf("[recv] Not a frame \n");
		}
	}
}
```

```
❱ ./server
[info] listening...
[info] client is accepted
[info] Waiting for frame 0
[recv] Frame 0
[info] Waiting for frame 1
[recv] Frame 1
[info] Waiting for frame 2
[recv] Expected 2 but got frame 0 discarding ...
[info] Waiting for frame 0
[recv] Frame 0
[info] Waiting for frame 1
[recv] Frame 1
[info] Waiting for frame 2
[recv] Frame 2
[info] Waiting for frame 3
[recv] Frame 3
[info] Waiting for frame 4
[recv] Frame 4
```

```
❱ ./client
[send] frame 0
[recv] recieved ACK : ACK 1
[send] frame 1
[recv] recieved ACK : ACK 2
[error] Expected ACK 3 but have ack 2
[send] frame 0
[recv] recieved ACK : ACK 0
[send] frame 0
[recv] recieved ACK : ACK 1
[send] frame 1
[recv] recieved ACK : ACK 2
[send] frame 2
[recv] recieved ACK : ACK 3
[send] frame 3
[recv] recieved ACK : ACK 4
[send] frame 4
[recv] recieved ACK : ACK 5
```