1. Bit Stuffing

1RV21EC174

Sunil Joshi

```
#include<stdio.h>
#include<string.h>
int main(){
char ch,array[100]="01111110",read_array[50];
int count=0,i=8,j,k;
printf("Enter data to be transmitted: ");
while((ch=getchar())!='\n'){
array[i++]=ch;
if(ch=='1'){
count++;
if(count==5){
array[i++]='0';
count=0;}}
else{
count=0;}}
strcpy(&array[i],"01111110");
printf("\nTransmitted bit stream (After stuffing) at the transmitter side is: %s\n",array);
j=strlen(array);
count=0;
k=0;
for(i=8;i<j-8;i++){
read_array[k++]=array[i];
if(array[i]=='1'){
count++;
if(count==5&&array[i+1]=='0'){
i++;
count=0;}}else{
count=0;}}
read_array[k]='\0';
printf("Destuffed data at the receiver is: %s\n",read_array);
return 0;}
```

2. Character Stuffing

```
#include<stdio.h>
#include<string.h>
#define DLE 16
#define STX 2
#define ETX 3
int main(){
char ch;
char arr[100]={DLE,STX};
int i=2,j;
printf("\nEnter the data stream (CTRL+B->STX, CTRL+C->ETX, CTRL+P->DLE):\n");
do{
scanf("%c",&ch);
if(ch=='\n') break;
if(ch==DLE){
arr[i++]=DLE;
} else if(ch==2){} else if(ch==3){} else {}
arr[i++]=ch;
}while(ch!='\n');
arr[i++]=DLE;
arr[i++]=ETX;
printf("\nThe stuffed stream is\n");
for(j=0;j<i;j++){
if(arr[j]==DLE){
printf("DLE");
} else if(arr[j]==STX){
printf("STX");
} else if(arr[j]==ETX){
printf("ETX");
} else {
printf("%c",arr[j]);}}
printf("\nThe de-stuffed data is\n");
for(j=2;j<i-2;j++){
if(arr[j]==DLE){
```

```
j++;
} else if(arr[j]==STX){} else if(arr[j]==ETX){} else {
printf("%c",arr[j]);}}
return 0;}
```

3. CRC

```
#include<stdio.h>
#define DEGREE 16
int mod2add(int,int);
int getnext(int*,int);
int result[30];
void calc_crc(int length){
int ccitt[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
int i=0,pos=0,newpos;
while(pos<length-DEGREE){
for(i=pos;i<pos+DEGREE+1;++i)result[i]=mod2add(result[i],ccitt[i-pos]);
newpos=getnext(result,pos);
if(newpos>pos+1)pos=newpos-1;
++pos;}}
int getnext(int array[],int pos){
int i=pos;
while(array[i]==0)++i;
return i;}
int mod2add(int x,int y){
return(x==y?0:1);}
int main(){
int array[30],length,i=0;
char ch;
printf("Enter the data (Message) stream:");
do{
scanf("%c",&ch);
if(ch=='\n')break;
array[i++]=ch-'0';
}while(ch!='\n');
length=i;
for(i=0;i<DEGREE;++i)array[i+length]=0;
length+=DEGREE;
for(i=0;i<length;++i)result[i]=array[i];
calc_crc(length);
printf("\nThe transmitted frame is:");
```

```
for(i=0;i<length-DEGREE;++i)printf("%d",array[i]);
for(i=length-DEGREE;i<length;++i)printf("%d",result[i]);
printf("\nEnter the stream for which CRC has to be checked:");
i=0;
do{
scanf("%c",&ch);
if(ch=='\n')break;
array[i++]=ch-'0';
}while(ch!='\n');
length=i;
for(i=0;i<length;i++)result[i]=array[i];
calc_crc(length);
printf("\nCalculated Checksum:");
for(i=length-DEGREE;i<length;i++)printf("%d",result[i]);
return 0;
}
```

4. Encryption using Substitution

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main(){
char seq[36]="qwertyuiopasdfghjklzxcvbnm1234567890";
char data[50];
char encoded[50];
int i,len;
printf("\nEnter data: ");
fgets(data, 50, stdin);
len=strlen(data);
if(data[len-1] == '\n') data[--len] = '\0'; // Remove the newline character
for(i=0;i<len;i++){
if(isupper(data[i]))
encoded[i]=seq[data[i]-'A'];
else if(islower(data[i]))
encoded[i]=toupper(seq[data[i]-'a']);
else if(isdigit(data[i]))
encoded[i]=seq[data[i]-'0'+26];
else
encoded[i]=data[i];}
encoded[len]='\0';
printf("\nEncoded string is: %s\n",encoded);
return 0;}
```

5. Decryption using Substitution

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main(){
char seq[36]="qwertyuiopasdfghjklzxcvbnm1234567890";
char data[100];
char decoded[100];
int i,j,len,present=0;
printf("\nEnter data: ");
fgets(data, 100, stdin);
len=strlen(data);
if(data[len-1] == '\n') data[--len] = '\0'; // Remove the newline character
for(i=0;i<len;i++){
for(j=0;j<36&& !present;++j){
if(seq[j]==tolower(data[i])){
if(isupper(data[i]))
decoded[i]='A'+j;
else if(islower(data[i]))
decoded[i]='a'+j;
else
decoded[i]='0'+(j-26);
present=1;}}
if(!present)
decoded[i]=data[i];
else
present=0;}
decoded[len]='\0';
printf("\nDecoded string is: %s\n",decoded);
return 0;}
```

6. Encryption using Transposition

```
#include<stdio.h>
#include<string.h>
int main(){
char data[100];
char wrd[]="MEGABUCK";
char cipher[20][8];
int seq[8];
int i,j,k,cnt;
for(i=0;i<strlen(wrd);i++){
cnt=0;
for(j=0;j<8;j++)
if(wrd[i]>wrd[j])
++cnt;
seq[i]=cnt;}
printf("\nEnter data: ");
fgets(data, 100, stdin);
cnt=strlen(data);
if(data[cnt-1] == '\n') data[--cnt] = '\0'; // Remove the newline character
for(i=0;i<cnt;i++)
cipher[i/strlen(wrd)][i%strlen(wrd)]=data[i];
if(i%strlen(wrd)!=0){
for(j=i%strlen(wrd);j<strlen(wrd);j++){
cipher[i/strlen(wrd)][j]='.';
cnt++;}}
printf("\nEncrypted data:\n");
for(i=0;i<8;i++){
for(j=0;j<8;j++)
if(seq[j]==i)break;
for(k=0;k<cnt/8||k==0;k++)
printf("%c",cipher[k][j]);}
printf("\n");
return 0;}
```

7. Decryption using Substitution

```
#include<stdio.h> #include<string.h>
int main(){
char data[100];
char wrd[]="MEGABUCK";
char cipher[20][8];
int seq[8];
int i,j,cnt,c;
for(i=0;i<strlen(wrd);i++){
cnt=0;
for(j=0;j<strlen(wrd);j++)
if(wrd[i]>wrd[j])
++cnt;
seq[i]=cnt;}
printf("\nEnter data: ");
fgets(data, 100, stdin);
cnt=strlen(data);
if(data[cnt-1] == '\n') data[--cnt] = '\0'; // Remove the newline character
if(cnt%strlen(wrd)!=0)
printf("\nError: Invalid Input\n");
else{
for(i=0;i<8;i++){
for(c=0;c<8;c++)
if(seq[c]==i)
break;
for(j=0;j<cnt/strlen(wrd);j++)
cipher[j][c]=data[i*(cnt/strlen(wrd))+j];}
for(j=0;j<strlen(wrd);j++){
if(cipher[cnt/strlen(wrd)-1][j]=='.')
cipher[cnt/strlen(wrd)][i%strlen(wrd)]=' ';}
printf("Decrypted data: ");
for(i=0;i<cnt;i++)
printf("%c",cipher[i/strlen(wrd)][i%strlen(wrd)]);
printf("\n");}
return 0;}
```

8. Kruskal

```
#include<stdio.h>
#include<string.h>
struct node{int set;} node[100];
struct edge{int first_node,second_node,selected,distance;} e[100];
int edge_count=0;
void getdata(int index,int total) {
int i;
for(i=index;i<total;i++) {
if(i!=index) {
printf("\nEnter distance between Vertex %c and %c:",index+65,i+65);
scanf("%d",&e[edge_count].distance);
e[edge_count].first_node=index;
e[edge_count].second_node=i;
++edge_count;}}}
void init(int total) {
int i;
for(i=0;i<total;i++) node[i].set=i;
for(i=0;i<edge_count;i++) e[i].selected=-1;}
void sort() {
int i,j;
struct edge temp;
for(i=0;i<edge_count-1;i++) {
for(j=0;j<edge_count-i-1;j++) {
if(e[j].distance>e[j+1].distance) {
temp=e[j];
e[j]=e[j+1];
e[j+1]=temp;}}}}
int main() {
int i,total,j,k,m,n,edgeselected=0,nodel,noder;
printf("\nEnter the number of nodes:");
scanf("%d",&total);
for(i=0;i<total;i++) getdata(i,total);
init(total);
sort();
```

```
printf("\nThe Sorted order of edges:");
for(i=0;i<edge_count;i++)
printf("\nedge: %d first node: %c second node: %c distance:
%d",i,e[i].first_node+65,e[i].second_node+65,e[i].distance);
i=0;
do {
e[i].selected=1;
nodel=e[i].first_node;
noder=e[i].second_node;
if(node[nodel].set==node[noder].set) e[i].selected=-1;
else {
edgeselected++;
m=node[nodel].set;
k=node[noder].set;
for(n=0;n<total;n++) if(node[n].set==k) node[n].set=m;}
i++;
} while(edgeselected<(total-1));
printf("\nMinimum Spanning Tree is:");
for(i=0;i<edge_count;++i) if(e[i].selected==1)
printf("\n%c<------->%c Distance %d",e[i].first_node+65,e[i].second_node+65,e[i].distance);
return 0;}
```

9. Prim's

```
#include<stdio.h>
#define infinity 999
int prime(int cost[10][10],int source,int n) {
int i,j,sum=0,visited[10],cmp[10],vertex[10];
int min,u,v;
for(i=1;i<=n;i++) {
vertex[i]=source;
visited[i]=0;
cmp[i]=cost[source][i];}
visited[source]=1;
for(i=1;i<=n-1;i++) {
min=infinity;
for(j=1;j<=n;j++)
if(!visited[j] && cmp[j]<min) {
min=cmp[j];
u=j;}
visited[u]=1;
sum=sum+cmp[u];
printf("\n %d-> %d sum=%d",vertex[u],u,cmp[u]);
for(v=1;v<=n;v++)
if(!visited[v] && cost[u][v]<cmp[v]) {
cmp[v]=cost[u][v];
vertex[v]=u;}}
return sum;}
void main() {
int a[10][10],n,i,j,m,source;
printf("\nEnter the number of vertices");
scanf("%d",&n);
printf("\nEnter the cost matrix: 0 self loop & 999 no edge\n");
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
scanf("%d",&a[i][j]);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
```

```
if(a[i][j]!=a[j][i]||(a[i][i]!=0)) {
printf("\nInvalid entry\nCost matrix should be symmetrical & the diagonal elements are zero");
return;}
printf("\nEnter the source:");
scanf("%d",&source);
m=prime(a,source,n);
printf("\n\nTotal cost=%d",m);}
```

10. Server

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
void error(const char *msg) {
    perror(msg);
    exit(1);}
int main(int argc, char *argv[]) {
    int sockfd, newsockfd, portno;
    socklen_t clilen;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;
    if (argc < 2) {
        fprintf(stderr,"You haven't provided port Number, please enter port number\n");
        exit(1);}
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) error("Server : error at port opening");
    bzero((char *) &serv_addr, sizeof(serv_addr));
    portno = atoi(argv[1]);
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);
    if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) error("Server : Error at binding");
    listen(sockfd,5);
    clilen = sizeof(cli_addr);
    newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
    if (newsockfd < 0) error("Server : Error while accepting");
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    if (n < 0) error("Server : ERROR reading from socket");
```

```
printf("MY message is : %s\n",buffer);
n = write(newsockfd,"I Have Received your message",30);
if (n < 0) error("Server : Error while writing to socket");
close(newsockfd);
close(sockfd);
return 0;}
```

11. Client

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
void error(const char *msg) {
    perror(msg);
    exit(0);}
int main(int argc, char *argv[]) {
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;
    char buffer[256];
    if (argc < 3) {
        fprintf(stderr,"usage %s Enter your hostname & port number\n", argv[0]);
        exit(0);}
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) error("Client : Error While opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"Client : Error, host not found\n");
        exit(0); }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr, (char *)&serv_addr.sin_addr.s_addr, server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) error("Client : Error while
connecting to server");
    printf("Please enter your message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
```

```
n = write(sockfd,buffer,strlen(buffer));
if (n < 0) error("Client : Error while writing to socket");
bzero(buffer,256);
n = read(sockfd,buffer,255);
if (n < 0) error("Client : Error while reading from socket");
printf("%s\n",buffer);
close(sockfd);
return 0;}
```

12. RSA

```
#include <stdio.h>
#include <math.h>
int gcd(int a, int b) {
    while (b != 0) {
        int t = b;
        b = a % b;
        a = t;}
    return a;}
int modExp(int base, int exp, int mod) {
    int result = 1;
    while (exp > 0) {
        if (exp % 2 == 1)
            result = (result * base) % mod;
        base = (base * base) % mod;
        exp /= 2;}
    return result;}
int modInverse(int e, int phi) {
    int t = 0, newT = 1;
    int r = phi, newR = e;
    while (newR != 0) {
        int quotient = r / newR;
        int tempT = t;
        t = newT;
        newT = tempT - quotient * newT;
        int tempR = r;
        r = newR;
        newR = tempR - quotient * newR;}
    if (r > 1)
        return -1;
    if (t < 0)
        t += phi;
    return t;}
int main() {
    int p, q, n, phi, e, d;
```

```
int plaintext, ciphertext, decryptedtext;
printf("Enter prime number p: ");
scanf("%d", &p);
printf("Enter prime number q: ");
scanf("%d", &q);
n = p * q;
phi = (p - 1) * (q - 1);
printf("Enter value for e (1 < e < %d): ", phi);
scanf("%d", &e);
if (gcd(e, phi) != 1) {
   printf("e must be coprime with φ\n");
   return 1; }
d = modInverse(e, phi);
if (d == -1) {
   printf("Modular inverse does not exist\n");
   return 1;}
printf("Enter plaintext integer to encrypt (0 < plaintext < %d): ", n);
scanf("%d", &plaintext);
ciphertext = modExp(plaintext, e, n);
printf("Ciphertext: %d\n", ciphertext);
decryptedtext = modExp(ciphertext, d, n);
printf("Decrypted plaintext: %d\n", decryptedtext);
return 0;}
```