```
#include <stdio.h>
int *in,**d1,**d2,x,y,**g,**d;
void create(int x,int y);
int staging(int x,int y);
int duplication(int x,int y);
int indexing(int x,int y3,int y);
void pimp(int x,int y3,int a);
void decode(int x,int y3);
void wxyz(int x,int y3);
main()
{
        int i,j,y1,y2,y3,a,q;
        printf("----------------------------------------------------------------------");
   printf("\n\n   Refer to Section 3.9 of Digital Principles and Applications");
   printf("\n          by Leach, Malvino and Saha, TMH, 2006");
   printf("\n\n----------------------------------------------------------------------");
        printf("\n Please give the number of variables you want to minimize - ");
        scanf("%d",&x);
        printf("\n\nIn this program your inputs are designated as : ");
        for(j=x-1;j>=0;j--)
                printf("a[%d]",j);
        printf("\n\nPlease give the number of minterms that you want to minimize - ");
        scanf("%d",&y);
        in=(int *)malloc(y * sizeof(int));
        d=d1=(int **)malloc(y * sizeof(int *));
        for(i=0;i<y;i++)
                d[i]=d1[i]=(int *)malloc((x+1)*sizeof(int));
        for(i=0;i<y;i++)
        {       printf("\nPlease give decimal indices of minterms one at a time : ");
        scanf("%d",&in[i]);}
        create(x,y);
   y1=y*(y+1)/2;
   d2=(int **)malloc(y1*sizeof(int *));
   for(i=0;i<y1;i++)
                d2[i]=(int *)malloc((x+1)*sizeof(int));
        y2=staging(x,y);
        y3=duplication(x,y2);
        a=indexing(x,y3,y);
        pimp(x,y3,a);
        printf("\n\nThe essential prime implicants giving minimized expression are:\n\n");
        decode(x,y3);
}

void create(int x,int y)
{
```

```
        int i,j,a;
        for(i=0;i<y;i++)
        {a=in[i];
                for(j=0;j<x;j++)
                {
                        d[i][j]=d1[i][j]=a%2;
                        a=a/2;}
                d[i][x]=d1[i][x]=8;
        }
}
int staging(int x,int y)
{
        int i1,j1,k1,t1,i2,j2,t2,c;
        i2=0;c=0;
        for(i1=0;i1<(y-1);i1++)
        {
                for(j1=i1+1;j1<y;j1++)
                {
                        t1=0;
        for(k1=0;k1<x;k1++)
                        {
                                if(d1[i1][k1]!=d1[j1][k1])
                                {
                                        t1++;
                                        t2=k1;
                                }
                        }
                                if(t1==1)
                                {
                                        for(j2=0;j2<t2;j2++)
                                                d2[i2][j2]=d1[i1][j2];
                                        d2[i2][t2]=3;
                                        for(j2=t2+1;j2<y;j2++)
                                                d2[i2][j2]=d1[i1][j2];
                                        d2[i2][x]=8;
                                        d1[i1][x]=9;
                                        d1[j1][x]=9;
                                   i2++;
        }}}
   for(i1=0;i1<y;i1++)
   {
                if(d1[i1][x]==8)
                {
                        for(j1=0;j1<=x;j1++)
                                d2[i2][j1]=d1[i1][j1];
                                i2++;
```

```
                }}
        for(j1=0;j1<x;j1++)
        {
                if(d1[0][j1]==d2[0][j1])
                        c++;
        }
        if(c<x)
        {
                d1=(int **)malloc(i2*sizeof(int *));
                for(i1=0;i1<i2;i1++)
                        d1[i1]=(int *)malloc((x+1)*sizeof(int));
                for(i1=0;i1<i2;i1++)
                {
                        for(j1=0;j1<=x;j1++)
                                d1[i1][j1]=d2[i1][j1];
                }
        staging(x,i2);
        }
        else
                return(i2);
}
int duplication(int x,int y)
{
        int i1,i2,j,c,t;
   t=0;
        for(i1=0;i1<(y-1);i1++)
        {
                for(i2=i1+1;i2<y;i2++)
                {
                        c=0;
                        for(j=0;j<x;j++)
                        {
                                if(d1[i1][j]==d1[i2][j])
                                        c++;
                        }
                        if(c==x)
                                d1[i2][x]=9;
        }}
        for(i1=0;i1<y;i1++)
        {
                if(d1[i1][x]==9)
                        t++;
        }

   i2=y-t;
        d2=(int **)malloc(i2*sizeof(int *));
```

```
        for(j=0;j<i2;j++)
                d2[j]=(int *)malloc((x+1)*sizeof(int));
        i2=0;
        for(i1=0;i1<y;i1++)
                if(d1[i1][x]==8)
                {
        for(j=0;j<=x;j++)
                                d2[i2][j]=d1[i1][j];
                        i2++;
                }
        return(i2);
}
int indexing(int x,int y3,int y)
{
        int i1,j,c1,i2,c,a;
        c=0;a=1;
        for(j=0;j<x;j++)
                if(d1[0][j]==3)c++;
   for(i1=0;i1<c;i1++)
                a=a*2;
   g=(int **)malloc(y3*sizeof(int *));
        for(j=0;j<y3;j++)
                g[j]=(int *)malloc(a*sizeof(int));
        for(i1=0;i1<y3;i1++)
                for(j=0;j<a;j++)
                        g[i1][j]=-2;
                for(i1=0;i1<y3;i1++)
        {
                c=0;
                for(i2=0;i2<y;i2++)
                {
                        c1=0;
                        for(j=0;j<x;j++)
                        {
                                if((d2[i1][j]==d[i2][j])||(d2[i1][j]==3))
                                        c1++;
                                if(c1==x)
                                {
                                        g[i1][c]=in[i2];
                                        c++;
        }}}}
        return(a);
}
void pimp(int x,int y3,int a)
{
        int i1,i2,j1,j2,c,w,j3,c1,c2,j4,c3,c4;
```

```
c=0;
for(i1=0;i1<y3;i1++)
{
	for(j1=0;j1<a;j1++)
	{
		if(g[i1][j1]!=-2)
		{for(i2=0;i2<y3;i2++)
		{
			if(i2!=i1)
			{
				for(j2=0;j2<a;j2++)
					if(g[i1][j1]!=g[i2][j2])
						c++;
		}}
		if(c==a*(y3-1))
		d2[i1][x]=91;c=0;
}}}
for(i1=0;i1<y3;i1++)
{
	if(d2[i1][x]==91)
	{
		for(j1=0;j1<a;j1++)
		{
			if(g[i1][j1]!=-2)
			{
				for(i2=0;i2<y3;i2++)
				{
					if(i1!=i2)
					{
						for(j2=0;j2<a;j2++)
							if(g[i1][j1]==g[i2][j2])
								g[i2][j2]=-3;
}}}}}}
for(i1=0;i1<y3;i1++)
{
	if(d2[i1][x]==91)
	{
		for(j1=0;j1<a;j1++)
			if(g[i1][j1]!=-2)g[i1][j1]=-1;
}}
for(i1=0;i1<y3;i1++)
{
	if(d2[i1][x]!=91)
	{
		for(j1=0;j1<a;j1++)
		{
```

```
                    if(g[i1][j1]>=0)
                    {
                    for(i2=0;i2<y3;i2++)
                        if(i2!=i1)
                        {
                            for(j2=0;j2<a;j2++)
                            {
                                if(g[i2][j2]>=0)
                                {
                                    if(g[i1][j1]==g[i2][j2])
                                {
                                    w=i2;
                                if((d2[w][x]==90)||(d2[w][x]==8))
                                {
                                    for(j3=0,c1=0;j3<x;j3++)
                                        if(d2[i1][j3]==3)
                                            c1++;
        for(j3=0,c2=0;j3<x;j3++)
                                        if(d2[i2][j3]==3)
                                            c2++;
        if(c1>c2)
                                    {

                                        d2[i1][x]=90;
          g[i2][j2]=-1;
                                    }
                                  if(c2>c1)
                                    {
                                        d2[i1][x]=8;
                                        d2[i2][x]=90;
                                        g[i1][j1]=-1;
                                    }
                                    if(c2==c1)
                                    {

    for(j3=0,c3=0,c4=0;j3<a;j3++)
                                        {
                                            if(g[i1][j3]==-
1)c3++;
                    if(g[i2][j3]==-1)c4++;
                                        }
                                        if(c3>c4)
                                        {

    d2[i2][x]=90;d1[i1][x]=8;
                                           g[i1][j1]=-1;
```

```
                                                                        }
                                                                        if(c3==c4)
                                                                        {
                                                                                d2[i1][x]=90;
                                                                                g[i2][j2]=-1;
                                                                        }
                                                                        if(c3<c4)
                                                                        {
                                                                                d2[i1][x]=90;
                                        g[i2][j2]=-1;
                                                                }}}

                                                                if(d2[w][x]==91)
                                                                        d1[w][x]=8;
        }}}}}}}}
        return;
}

void decode(int x,int y3)
{
        int i,j;
        for(i=0;i<y3;i++)
        {
                if((d2[i][x]==91)||(d2[i][x]==90))
                {
                        for(j=x-1;j>=0;j--)
                        {
                                if(d2[i][j]==0)printf("a[%d]'",j);
                                if(d2[i][j]==1)printf("a[%d]",j);
                }}
   printf("\n\n");
        }
        return;
}
```