

Procedures Fib and Pref in C

Procedure Fib

```
#include <stdio.h>
#include <stdlib.h>
int n, k, p, b[100], s[100], nb=1;

void type()
{ int i;
  printf("%d ",nb++);
  for (i=1; i<=n; i++) printf("%d",b[i]);
  printf("\n");
}

void fib(int m, int j)
{ int i;
  if(j>0 && m>1 && m!=2*j-1)
  { if(b[m+1]==1) fib(m-1, j);
    else
    { if(b[m]==0)
      { fib(m-1,j);
        b[m]=1; b[s[j]]=0;
        s[j]= m;
        type();
        fib(m-1,j-1);
      }
      else
      { fib(m-1,j-1);
        b[m]=0;
        b[s[j-1]+2]=1;
        s[j]=s[j-1]+2;
        type();
        fib(m-1,j);
      }
    }
  }
}

int main()
{ int i,j=1;
  printf(" ENTER n: "); scanf("%d", &n);
  printf(" ENTER k: "); scanf("%d", &k);

  if(n<2*k-1) printf("ERROR");
  else { s[0]=-1; b[n+1]=0;
        for(i=1;i<=n;i++)
```

```

        { printf(" ENTER b[%d] ",i); scanf("%d", &b[i]);
          if(b[i]==1){ s[j]=i; j++; }
        }
        type();
        fib(n, k);
    }
}

```

 Procedure Pref

```

-----
#include <stdio.h>
int n, k, p, b[100],s[100], nb=1;
#define MAX(a,b) (((a)>(b))?(a):(b))

void type(int u, int v)
{ int i;
  printf("%d  ",nb++);
  for (i=1; i<=n; i++) printf("%d",b[i]);
  printf(",%d %d<- transposed bits",u,v);
  printf("\n");
}

```

```

void gen(int m, int j)
{ int u;
  if ( m!=j && j>0)
  { if(b[m]==0)
    { gen(m-1,j);
      u=s[j]; b[m]=1; b[s[j]]=0; s[j]=m;
      type(m,u);
      gen(m-1,j-1);
    }
    else
    { gen(m-1,j-1);
      s[j]=MAX(s[j-1]+1,(p+1)*j);
      if(s[j]!=m)
      { b[m]=0; ; b[s[j]]=1;
        type(m,s[j]);
        gen(m-1,j);
      }
    }
  }
}

```

```

int main()
{ int i,j=1;
  printf(" ENTER n: "); scanf("%d", &n);
  printf(" ENTER k: "); scanf("%d", &k);
  printf(" ENTER p (for p=0 combinations are produced, ..): "); scanf("%d", &p);

```

```

if(n<(p+1)*k) printf("ERROR");
else { for(i=1;i<=n;i++) b[i]=0;
      for(i=1;i<=n;i++)
      { printf(" ENTER b[%d] ",i); scanf("%d", &b[i]);
        if(b[i]==1){ s[j]=i; j++; }
      }
      printf("%d  ",nb++);
      for (i=1; i<=n; i++) printf("%d",b[i]);
      printf("\n");
      gen(n, k);
    }
}

```