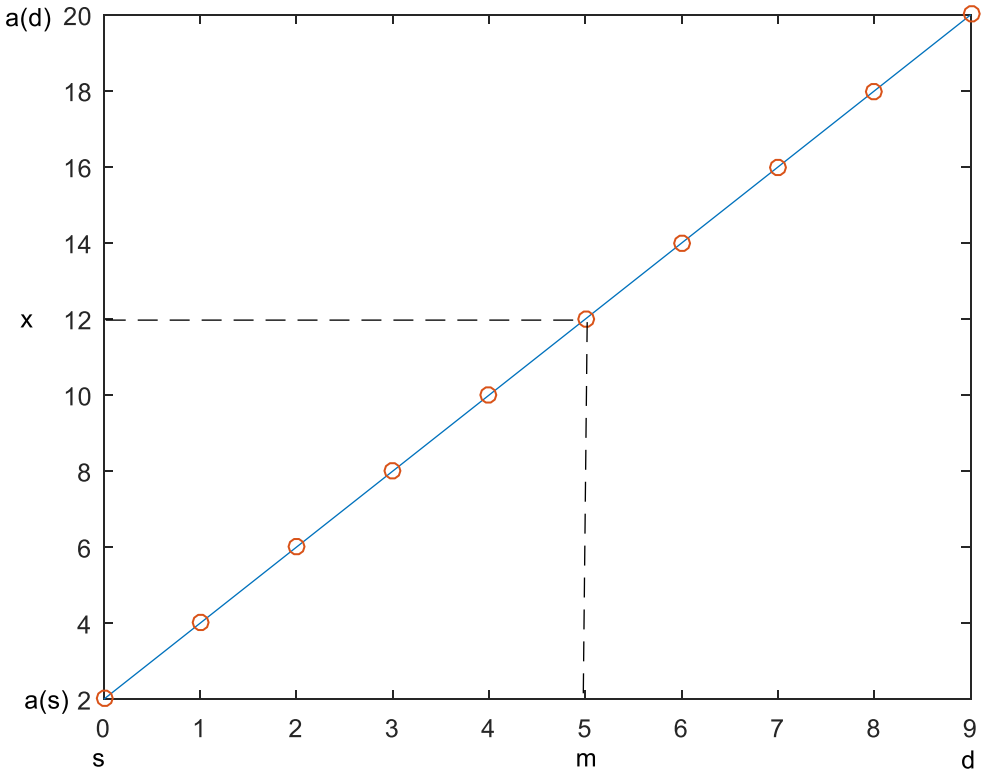

Interpolare

//(m-s)/(d-s)=(x-a(s))/(a(d)-a(s))

//m=s+(x-a(s))(d-s)/(a(d)-a(s))

```
#include <stdio.h>
void afisare(int t[],int N){
   int i;
   for(i=0;i<N;i++)
      printf("%d ",t[i]);
   printf("\n");
}
int cautare_liniara(int a[],int N, int x){
   int i=0;
   while(a[i]!=x&&i<N)
```

```
    i++;
   return i;
}
int tehnica_fanion(int a[],int N, int x){
   int i=0;
   a[N]=x;
   while(a[i]!=x)
    i++;
   return i;
}
int cautare_binara(int a[],int N, int x){
   int s,d,m;
   s=0;d=N-1;
   do{
      m=(s+d)/2;
      if(x>a[m]) s=m+1;
      else d=m-1;
   }while(a[m]!=x&&s<=d);
   return m;
}
int cautare_binara_performanta(int a[],int N, int x){
   int s,d,m;
   s=0;d=N-1;
   do{
      m=(s+d)/2;
      if(x>a[m]) s=m+1;
      else d=m;
   }while(s<d);
   return m;
```

```
}
int cautare_interpolare(int a[],int N, int x){
   int s,d,m=-1;
   s=0;d=N-1;
   if(x>=a[s]&&x<=a[d]){
   do{
      m=s+(x-a[s])*(d-s)/(a[d]-a[s]);
      if(x>a[m]) s=m+1;
      else d=m-1;
   }while(a[m]!=x&&s<d&&x<=a[d]&&x>=a[s]&&a[d]!=a[s]);
}
   return m;
}
void meniu(void){
   int op,x,index;
   int t[]={2 ,4,17,21,35, 55,66,67,89,99,-1};
   int N=10;
   do{
      printf("1. Cautare liniara\n");
      printf("2. Tehnica fanionului\n");
      printf("3. Cautare binara\n");
      printf("4. Cautare binara performanta\n");
      printf("5. Cautare prin interpolare\n");
      printf("6. IESIRE\n");
      afisare(t,N);
      printf("Elementul de căutat:");
      scanf("%d",&x);
      printf("Optiunea:");
      scanf("%d",&op);
```

```
    switch(op){
        case 1: index=cautare_liniara(t,N,x); if(index==N)printf("\n....\n"); else printf("\nElementul %d se regaseste pe pozitia %d\n",x,index);break;
        case 2: index=tehnica_fanion(t,N,x); if(index==N)printf("\n....\n"); else printf("\nElementul %d se regaseste pe pozitia %d\n",x,index);break;
        case 3: index=cautare_binara(t,N,x); if(t[index]!=x)printf("\n....\n"); else printf("\nElementul %d se regaseste pe pozitia %d\n",x,index);break;
        case 4: index=cautare_binara_performanta(t,N,x); if(t[index]!=x)printf("...."); else printf("\nElementul %d se regaseste pe pozitia %d\n",x,index);break;
        case 5: index=cautare_interpolare(t,N,x);if(index!=-1 && t[index]==x)printf("\nElementul %d se regaseste pe pozitia %d\n",x,index); else printf("\n...\n");break;
    }
  }while(op!=6);
}
int main()
{
  meniu();
return 0;
}
```