Submitted By

Name: Likhan Sarker

ID: 191-15-12958

```
coin-changing-GA-1.
# include LStdio.h>
 void sort (intara[], inta)
  Pint injes
     for (i=0; iLn; i++)
         1 for (j=0; j Ln-1-i; j++)
             2: f (anati)> anati-1])
                3 p=arati+1];
                  anatiti]= anati]i
        3 } } ana[i]=a
  void coin-change (int coins[), intn, intm)
    int ent [n]iii
    for (i=0: i Ln; i++) cont [i]=0;
     for ( i=n-1; 17=0; i++)
       { if (coins [i] L=m)
          < ent [i] &=my coins [i];
             m=m/, coins[i]s
```

prients (" change is not possible \n");

```
eisa
 Sprints (" coin need: In");
    for (i=n-1 i i=0 ; i--)
      } if (cn+ [i]!=0)
       prents ( " /d coin: /d times \n", coins[i]
                   , ent til) i
 int main ()
    int n=4, change = 15;
     int coins[] = {1,7,7,10};
     Sort (coins in);
     coin-chang (coins, or, change);
      returno;
```

Fractional-Knapsack-GA-1.C

```
#include LStdio.h>
  int man (inta; int b) } return (a7b)? a:b;}
  int: knapsack (intw. int wt[], intv[], intv[]
sortint i, wi.
     int K[n+1][w+1];
    for (i=0; i L=n; i++) {
       for (w=0; w==w; w++)}
          if (i==011w==0)
            K TIDEWJ = 0;
        else if (w+[i-1] L= w)
         KTIDEWD = mar(V[i-1]+K[i-1)
         [w-w+[:-1]], K[:-1][w]);
            KEIJEWJ=KEI-1JEWJ;
         2150
       rations K[n] [w];
      int main ()
```

```
Fractional - Knapsack - DA-1.C
```

```
#include LStdio.h>
int max (inta, intb) { return (a>b)? a: b;}
int knapsack (intw, intw[], intv[], intn)
 I inti, w;
    int K [n+1] [w+1) i
  Sor (i=0; i = n; i++) }
       (0=w110==i)ti
           KTI)[W]=0i
      else if (ut [i-1] L=w)
     K[i][w] = max(v[i-1]+K[i-1][w-w+
                [1-1]], K[1-1][W]);
          KEIJ [W] = K[i-1] [W]
   return k[n][w];
    int main ()
  ? in+v[)= \12,10,20,15};
     int wt [)={2,1,3,23;
    int n = size of (v)/size of (v[0));

prints ("Maximum protit: "/d", knapsack(w,
rutorino;
```

fibonacci - num - DA-1.C

```
#include Lstdio.h>
 int fib (intr)
 return ni
   return fib (n-1) + fib (n-2);
  int main()
 fint main ()
   in tais
   printf ("Enter Any number:");
   Printf ("Fibonacci number: 1.d", fib(n));
    scarf ("/d", &n);
    getchar();
    returnoi.
```

```
Frantional-Knapsack-DA-2.C
#include LStdio.h>
int max(inta, intb) { return (a76)? a, b;}
int knapsack (int w, int w/[], int v[], int n)
 fintini;
   int K[n+1)[w+1];
    for (1=0;14=n;1++)}
          if (1==0 11 w==0)
         WEIJEW) =0;
      else if (w+[i-1] L=w)
       K[i][w] = * max(v[i-1] + K[i-1][w-
              w+ti-1]], k [i-1][w]);
          KEIDENJ=KEI-1]EN];
     rietorn k[n][w];
 I int main ()
  fint V[)={20,10,307;
  int wt[) = {100,50,150};
  int n = size of (v)/size of (v[0));
  printf ("Maximum profit: 1/d", Knapsack
   retarno; (w, w+, v,n));
```

coin-changing-b1A-2.

```
#include Lstdio.h>
 void sort (int ara [], int n)
  tint i,i,Pi
   for (i=0)iLn;i++)
        Sfor (j=0;j∠n-1-1;j++)
            ?if(arati) > ara[i-1])
             { p = ara [ ;+1] ;
               arati+1) = arati];
               anati)=Pi
roid coin-change (int coins[], intm, intm)
   for (i=0; i Ln; i++) ent [i]=0;
  fint cont Enjis
    -for(i=n-1;i7=0;i--)
      ? if (coins [i] L=m)
           { ent [i]=m/coins [i]
            m=m'/. coins[i];
```

```
if (m!=0)
    prints ("change is not possible In");
elsa
    prient f ("coind need: \n");
      for (i=n-1;i>=0;i++)
     ?if(ent ti)!=0)
      grientf ("1.d coin: 1.d times In",
        e ainstil, enttill);
 int main ()
  fint ma n=5, charge = 12;
    int coins[]={2,5,3,4,6};
     sout (coins, n);
     coin-change (coins, n, change);
     raturn oi
```

Fibonacci - num - DA -2. L

```
#include LHdioh)
 int fib (int n)
  fint f[n+2) 11)
    f[0] = 0;
    f[1]=1;
    for (i=2; i L=n; i++){
       fti)=fti-1]+fti-2];
   7 Futuren of [n];
     int main ()
   fint n, ti
  printf ("Test case:");
   Scanf ("1.d", &+);
    for (inti=1; 12=+; i++) }
   print (" Number 1.d", i);
    scanf ("1.d", &n);
    printf ("Fibonacci /d:o/d \n", i, fibon);
     ruturen os
```