coin-changing-DA-1.

```
# include L Stdio.h7
 void coin_ change (int coin)[], int total coin, int change)
int m [change +1], minimum, i, j;
    m[0]=0;
  for (i=1; i <= change to; i++) {
      minimum = change +1;
  for (j=0; j L total coin; j++) {
if (coin [j] L=i) {
            if (m[i-coin[j]]+1 / minimum)
             minimum = m [i-coin [j]]+1;
          m[i] = minimum;
    if (m [change] = = 0)
    Preints ("change is not possible In");
       Prient f ("coin need: %d \n', m [change]);
   else
   Int main () {
   int i, total coin = 4, Change = 16;
    int coin [] = {1,2,8,123;
    coin - change (coin, total coin, change);
  > return o;
```

coin-changing-DA-2.C

```
#include LStdio.h7
void coin - chang (int coin[], int total coin, int charge)
  fint m[change +1], minimum, i, j;
   for ( i=1; i L = change; i++) {
       minimum = change + 1;
     for (j=0; j L total coin; j++) {
             if (m[i-coin[j]]+1 L minimum)
       if (coin[i) L=i){
            minimum = m[i-coin[j]]+1;
        } m[i]= minimum;
    if (m [change] == 0)
       prients ("charge is not possible in");
        printf ("coinneed: /d \n", m [change]);
     int main () {
    inti, total coin=3, chang=16;
    int coin [3] = { 1,5,10 };
    coin-change (coin, total coin, change);
     return oi
```

coin-changing-GA-1.C

include LStdio.h> void sort (intara[], inta) Pint i, j, P; for (i=0; izn; i++) 3 for (j=0; j Ln-1-i; j++) ? if (anati) > anati-1]) of p=arrati+1]; anati+1)=anati)i anati]=Pi void coin-change (int coins[), intn, intm) % int cnt [n]; i;

for (i=0; i∠n; i++) cn+ [i]=0;

for (i=n-1; i7=0; i++)

? if (coins [i] ==m) < ent [i] &=my coins [i]; m=mº/, coins[i]s

Prients ("change is not possible In");

0150 & prients (" coin need: In"); for (i=n-1 i i=0 ; i--) (0=1. [1] (cn+ [i] 1=0) prents (" /d coin: /d times \n", coins[] icizi, enttiz); int main () in+ n=4, change = 15; int coins[] = {1,7, 7, 10}; Sort (coins in); coin-chang (coins, or, change); return o;

coin-changing-b1A-2.

```
# include L5+dio.h7
 void sort (int ara [], int n)
  fint i,j, P3 bombons 1 Home
   for (i=0;iLn;i++)
        Sfor (j=0;j~n-1-i;j++)
           ?if(arati) > arra[i-1])
            1 p = ana [i+1];
               aratiti) = aratil;
               anati) = pi
roid coin-change (int coins[], intn, intm)
   for (i=0;izn;i++) ent [i]=0;
 fint ent [m], i;
    for(i=n-1;i7=0;i--)
      } if (coins [i] L=m)
          i ent [i]=m/coins [i]

m=m'/. coins [i];
```

```
if (m; =0)
   Prients ("change is not possible In");
 else
  Prient f ("coind need: \n");
     for (i=n-1;i)=0;i++)
  Sif(ent [i)!=0)
     prient f ("1.d coin: 1.d times In')
  2 ? coins[i], cnt[i]);
int main ()
 Sint ma n=5, change = 12;
   int coins[]={2,5,3,4,6};
    sort (coins, n);
    coin-change (coins, n, change);
   ruturn o;
```

fibonacci-num-DA-1.C

#include Lstdio.h) int fib (intr) 3 if (n L=1) return ni return fib (n-1) + fib (n-2); int main() fint main () int mi printf ("Enter Any number:"); Printf ("Fibonacci number: 1/d", fib(n)); scarf ("/d", &n); getcharl); returno;

Fibonacci - num - DA - 2. C

```
#include Lstdio.h)
 int fib (int n)
  3 int f[n+2], ii
    f [0] = 0;
     f[1]=1;
     for(i=2;iL=n;i++){
         fti)=fti-1)+fti-2];
   2 rutorn f[n];
     int main ()
  printf ("Test case:");
Scanf ("%d", &+);
for (intizzii L=+; i++) {
   Print ("Number 1.d", i);
    printf ("Fibonacii /d: o/d \n", i, fib(n));
      return os
```

Fractional - Knapsack - DA-1.C

```
#include LStdio.h>
int max (inta, intb) { return (a>b)? a: b;}
int knapsack (intw, intw[], intv[], intr)
 inti, wi
    int K [n+1] [w+1);
  Sor (i=0; i = n; i++) {
       if ( i = = 0 11 w = 0 )
          K[i][w]=0;
     else if (w+[i-1] L=w)
     kti)[w] = max(v[i-1]+k[i-1][w-w+
                 [i-1]], K[i-1][W]);
     else KEIJ [W] = KTi-1) [w]
    return k[n][w];
  } in+v[)= {12,10,20,15};
     int wt [)={2,1,3,23;
     int n = size of (v) / size of (v[0));

prints ("Maximum probit: "/d", knapsack(w, retorno;
```

Frantional-Knapsack-DA-2.C

```
#include LStdio.h)
int max (inta, int b) { return (a76) ? a, b; }
int knapsack (int w, int w/[], int v[], int n)
 & inti, wi
   in+ k[n+1)[w+1];
   for (i=0; i 4=n; i++)}
         if (1==0 11 w==0)
        いてiフィw) =0;
     else if (w+ [i-1] L=w)
K[i)[w] = * man(v[i-1]+K[i-1][w-
             w+ti-1]], k ti-1][w]);
         KCIJEWJ=KCi-1][W];
     return k[n][w];
   int main()
 fint V[]={20,10,307;
  int wt[) = {100,50,150};
  int n=sizeof(v)/sizeof(v[o));
  int w = 50;
  prients ("Maximum profit: 1.d", Knapsack
   retarno; (w, w+, v, n));
```

Fractional - Knapsack - GA-1.C

count aning though

int main ()

Hinclude LStatoshy (21-04.00) = (] V+mi int max (inta, int b) } return (a76)? a: b;} int knapsack (intw. int wt [), int vE), int n) int K[nt][witi]; for (i=0; i L=n; i++) { for(w=0; w=+)} if (1==011W==0) KTIDEWJ=0; else if (w+[i-1] (=w) Kti)[w] = man(v[i-1]+K[i-1) [w-w+ti-1]], kti-1][w]); KTIJTW)=Kti-1JTWZ; 2150 return K[n] [w];

Mehedi Hasan Munna 191-15-12946

Fractional-Knapsack +MA-116} in+V[]={30,40,45, 77,90} is showith int wit [] = {5,10,15,22,25} int n = size of (v) / size of (v[0)); print ("Maximum probit: 10d", knar sack (w, w+, v, n)); 100 11-01 10 of return (0) - Has in es will and (0==w110==1)+1 in concine Clso it (w) [1-1] (= w) -ilx+[1-i]v)ram = ConjCija [w-w+[i-1]] KEI-1][w] ことのアレー・コンニにいている i Custail another

Meledi Hearn v