```
Problem#1
                        +13/19/2006/194
 Test care+1
  Ti= a=4, b=7, c=6, d=1
  pa = 4
line(6)
pargett) = Object of A
pb-) get () = object of B
   4>1
  Pa=1
  Pb= 4
line(13)
  PC = 6
line (14)
 panget()= object of B
  pc - get () = object of c
   1>6 (false)
line(18)
  parget (1 = Object of B
  pc - get() = object of c
  pb + get () = Object of A
   1+6 >=4 (True)
 Test case # 2
```

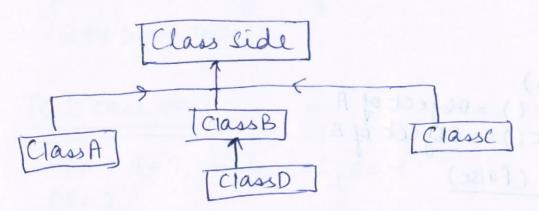
T2 = 9 = 3, b = 3, c = 1, d = -1

```
pa = 3
    Pb=
  line(6)
  pa - gel- () = Object of A
 pb-)get()=ObjectogoB
   3)
   Da = 1
   pb=3
line(13)
   PC=O
lul (4)
  pa=0, pa→get()=Object of B
pc=1 pc→get()=Object of D
line (18)
 pb-1 get() = Object of A
   0+1>=3 = false
 Test case# 3
   T_3 = Q = 3, b = 3, C = 4, d = -1
  line (6)
   pa - get () = Object of A
  pb + get() = Object of B
    3>4= False
 line 13
     PC = 4
```

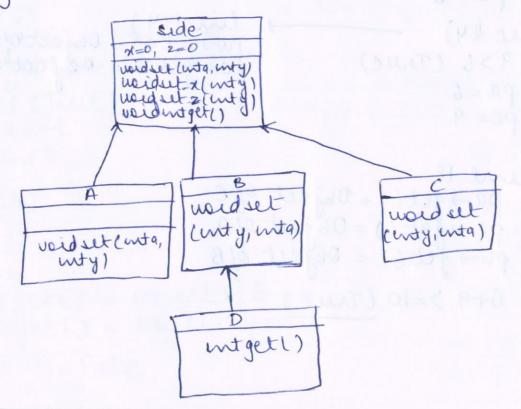
```
line(14)
   parget () = Object of A
   pb-) get() = Object of B
     374-false
    line (18)
   panguti = Object of A
   pc = get() = Object of D
   pb-) get() = Object ofB
    3+4)=4 (True)
 Test case #4
  Ty = 0 = 7, b = 4, c = 6, d = -1
   pa = 7
   Pb= -1
 line (6)
  pa - get() = Object of A
  pb-yet() = Object of B
   7>1 (true)
     Pb = 7
pene (13)
  PC= 84
line 14
  panget() = Object of B
   pc-jget() = Object of D
   -1>4 = false
```

```
leve 18
    pa + get-() = Object- of B
    pc > get() = Object of D
    pb- geti) = object of A
     1+4)=7 (false)
   Test case #5
 T_5 = a = 9, b = 5, c = 6, d = 10
 pa = 9
 pb= 10
 lune (6)
pa-iget () = Object of A
 phoget () = Object of B
   9>10 (false)
  leve (3)
   PC= B
                         line (14)
pa-get() = Object of A
line (4)
                         pc=get() = object of c
   9>6 (True)
   pa = 6
   PC = 9
 line 18
    pa - get () = Object of C
    penget () = Object of A
   pb + get () = Object of B
    6+9 >=10 (True)
```

Test case#1 = a=4, b=7, c=6, d=1Test case#2 = a=3, b=3 c=1, d=-1Test case#3 = a=3, b=3, c=4, d=-1Test case#4 = a=7, b=4, c=6, d=-1Test case#5 = a=9, b=5, c=6, d=10



Polymorp hic calls at line number = 6, 14, 18



## Problem# 2

- 1. int search(ent n, mtx, mtd)

  { mt i, flag, z;

  2. i=1;
- 3. flag = -1;
- 4,5 y(n>0) & + (a[i] == x)) flag = 1;
- 6 while ((ixn) 47 ! [flag==1){
- 7,8 if (a[i] = = x) flag = =1;
- 9 i++;
- 10,11 if (n<=0) & & (flag >=1)) retuen 1; 12 else retuen 0;

Condition al (4)

((no <0) | [ (ao [ ]] = xo)) 27 ((no >1) 22 [H==1) 22 (90[1]!= No) &x (no <211[]==1)) +x (no >0) +4 (1 >1) needs to be true. But here (1) is fale as

-1 is less them I which makes the whole
condition is false thence, we cannot proceed further

Condition alts)

(no <0) | (ao(i) |= no)) } { (no >1) } { (1 == 1) } } } ( Qo[] = xo) & P((no>2) & P((-1==1)) A P

ao[2)!= 20) & [(no =3) | [(1==1)] & & (no <0) 28(1821)) needs to be true.

But, here (121) is false as I is less then 1 which makes the whole condition false. Hence, we cannot execute beanch (10,11),

condition at (6)

((no <0) // (ao(1)!= xo)) ++((no>1) ++(H==1)) +7 (ao[1)!=no) & & ((no)2) & & !(1==1)) & & (ao[2)==no) & & (no)3) & & !(1==1) needs to be true.

But here! (1==1) is falseas (1==1) yields bul but negation makes it falst. Hence, the conditionfully is false twe cannot

peould further. (1) do millowed Condition at (7) Soll ) 88/08 = 01000 xx100000 ((no <0) 11 90[17!= No)) ++ ((no ono>1) 27! (+==1)) 27 ((no >2) 22 1 (1==1)) 27 (90[2] == no) 27 ((no 63)1) (1==1)) & P((no <0) & P(121)) needs to be true. But here (no>2) & f (no <0) is a contradiction which makes the full condition false hence we can't execute (10,11) branch.

Condution at (8)

((no <0) | (aol1) != xo)) 27 ((no <1) | (+==1)) 27 ((no≤0) 47(121)) needs to be true.

But, here in this 121 is false hence, the whole condition is false hence we cannot proceed further.

condition at (b)

for condition at live 10, n is not changing & flag = 1 will not change foreuer In boop n will vory so line 10 will never be true

condition at (C) relation at (C)
Flag is 1 co condition at line 10 will neuer be true

condition at (1) ((no)0)42 (ao(1)== no)27 (no)1)27 !(1==1)) Since |== | is-bue, !(1==1) is false which makes the whole condition is false hence we Cannot proceed further from line 7 Condition at (2) ((nos0) 27 (aol1) == 20)) 27 ((12 no) 11 (1==1) 47 ((no≤0) ff(1≥1)). If no so then no ≤0 cannot hold tell, creating a contradiction at line 11, therefore, no off no so is a contradiction and beauch (10,11) is non executable. condition at (3) ((no 60) 11 (ao(1) != no)) 27 ((no )1) 27! (+==1))42 (auli) == No) => (no 40) 27 (no>1) 27 ! (+=1) 24 ao(1) == no

has to betwee OR a of 1]! = no) le (no) 1) ff 1(+==1) le (ao[1] = = no) has to be true. = no <0 ff no >1 is contradicting meaning we consider (ao[] != No) 47 (no) 1) 47 (4==1) AP(ao[1] = = xo). Hence, (aoli) = xo) + f(aoli) == xo) is contradicting hence we cannot proceed further

froblem#3 loop Invariant Pradicate 6 for all (i+1 \le t \le n): min \le [a[t]] & K = nog time execution is at predicated coop entry Path Path = 1,2,3,4,6 i=n+, min=a[n] for all (i+1 \le t \le n)! min \le |a[t]| forall (n2t2n): min < [a[t]] forall (n 2t > not + ) ! min < |a[t] a[n] > a[n] is substituting for min = (a[n] +t=n) path = 1, 2, 3, 4, 5, 6 min = -a(n), i = n - 1for all (i+1 < t < n)! min < | a(t) foeall (n+1 \le t \le n): min \le |a[t] foeall (net 2n): min & (act] =) re-a[n] = |a[n]| =) terre is substituted in place of min = -a[n] ft=n Sume some value (K) loop meariant holds true

```
forall (ik+1 \le t \le n): mink \le |a[t]|
```

conditions

3) Proung for k+1 Path = 6,7,8,11,6

7= [([aix] 20] ++ (minx >(a[ix])] =) true

8 = minkt1 = a[ik]

11= 10 ck+1= ck+positedudus a [N] of [CM]

6 = iktl = ikt

mink+1 = alik]

forall & [ix+1+1 & t & n]: mink+1 & |a(t)| forall[ix-1+1 \le t \le n]: mink+1 \le | act]

forall [ix+1 =t =n]: mink+1 = |a[t] | 47 mink+1 =

a[ik] 2027 mink+1 = a[ik] =):, |a[ik] = |a[ik] =) toue

forall (ix+1 <t < n): mink+1 < act]

(i) mink+1 < mink =) true

(ii) forall (ix+1 \le t\le n): min ko \le [a[t]]
forall (ix+1 \le t\le n): min k+1 \le [a[t]]

Path= 6,7,0,9,10,11,6

1= [[a[ik] 20] ff (mink > a[ik])

9= [a[ik] < 0] 27 (mink > - 9[ik])

 $10 = \min_{k+1} = -9[ik]$ 

 $N = i_k + 1 = i_k - 1$ 

 $G = i_{k+1} = i_k + i_{k+1}$   $min_{k+1} = -a[i_k]$ 

=) [(a[ik]<0] | [mink = a[ik]) 47 [a[ik) <0] 47 (mink > -a [ik]))

=) [[a[ik] <0] & f [mink > -a[ik]]) // [[minkealik]] \* + [a [ik] <0] + + [mink) -a [ik]]

=) mink ) | a[ik]

=) mink > mink+1

```
foeall (ix+1 < t≤n):minx+1 ≤ a[t]/
  forall (ix++1 &t &n): min_K+1 & [a[+]]
 forall (ix+1+ 5 t 5 h)! min x+1 = [a [t]]
 forall (ix+1 \( t \le n \): min \( \text{p1} \le | a[t] | & P
                     min & | a Cited of minte
  min Kt1 = a [ik] < mink
    -a Cir] = |a Cir] = true
foeall (ixt1 \le t \le n): min x+1 \le |a(t)|
 (i) minkt, < mink => true
(ii) forall (ik+1 < t < n): mink < [act]
forall lix+1 \(\delta\text{t} \) 1 min \(\kappa\text{t} \le 1 act)
 Path = 6,7,89,11,6
     ![(a[ik] >0) & + (mink) a(ik))]
       ! [[a[in] <0) & & (mink) - a (ix))]
       ix+1 = ix+
      UK+1 = ix-
 min kt1 = mink
```

```
= 1. C(a[ik] 20] & + (mink > a[ik])) & + ! [(ablik) 

~ (a) & + (mink) - a[ik])

[a[ik] ~ (a[ik] ~ (a[ik])] & +

(a[ik] ~ 20) | 1 (mink & -a[ik])]

[a[ik] ~ (a[ik] ~ (a[ik]))
```

- =) (a[ik] <0) +4 (mink = -a [ik]) =) teue mink = |a[ik]|
- for all (ik+1+1 & t & n); min k+1 & a(t)

  for all (ik-1+1 & t & n); mink+1 & a(t)

  for all (ik+1 & t & n); mink+1 & a(t) & for all (ik+1 & t & n); mink+1 & a(t) & for in k+1

  for all (tadix & t & n); mink & (a(t)), loop invariant

  is wered on termination

forall (i+1 < t < n): min < |act ] |
forall (0+1 < t < n): min < |act ] |
forall (1< t < 6 n): min < |act ] |
forall (1< t < 6 n): min < |act ]