```
LAB 10 - Brogram 9 - Binonical Heap
 stourt Node
    int data, degree;
    Node *dild, sibling, sparent;
 list < Nocle x 7 insert (list < Nocle x) - head, int key)
      Node *temp = new Node ( key);
      return insert Abore In Heap (_head, temp);
 Node * get Min (lest < Node * 7 - heap)
 lutz node >>: iteratore it = _heap. legin();
  Node stemp = xit;
                                to preduce the
       while (it! = heap. end ())
          if ((xit)-)data < terry -> data)
                  temp = xit;
 The way i Etat glow 1 135 M 3 . From the appearance of
 list < nocle 47 insert A bree In Heap ( dist ( Avode x 7_heap, Node x ten).
       femp. push-back (thee);
      list < Node + > temp;
       temp = union Binomial Heap (-heap, temp).
        settin adjust (temp);
                                               Amusia
```

```
lint < Nocle x7 intract Min (list < Nocle x > heap)
         list crode &7 new-heap, lo;
          Node stemp:
          temp = getrlin (-heap);
          list < node > 7 :: ituator it;
      it = heap. legin();
            while (it != heap. end(1)
                  if (|xit != temp)
                        new heap. push back (*it);
                   Capacity = a stores tole ) with ty = draw
              lo = remove Min From Dree Return & Heap (temp);
              new heap = union sinonvial Heap ( new-heap, lo),
              new-heap = adjust (new-heap); // seavange, theheap
              return new heap;
                                             Il degree in the heap
      Noch * merge Binomial Trees (Node *bi, Noch *b2)
(b) data 7 b2 -> deta)
                   swap (bi, bz)
             bo -> parent = bi / final ca domes de
         bo -> parlent = 5 /
bo -> ribling = bi-> child;
              by -> wild = bz;
              51-degree ++; / sell i color assessiones
                                                   frume
```

में क्षेत्रकार विशेष व्यूपे विश्व स्थानिकी है

```
list Node * 7 union Binomial Heap (list ( Node * 7 li)
                                    list (Node =7 /2)
        list ( Mode => _new;
         list < Node * 7: iterator it = l. begin();
        lut ( Node x 7: ituator OF = lz. legin().
         while (it = liend(182 ot!= 12. end(1)
               if ((xit)->deque <=(xot)->deque)
                    -new. prush_back (xit);
                     new.push_back(200+);
                 elie ?
         while (it!= li. end())
              -new-push bach ( sit );
          while (of ! = 12. and (1)
               _new.push_back ( * ot );
          retur news,
```

3

Queru

Anune Margin's List (Node & > heap.)

Node & tere - wild;

Node & lo;

while (temp)

to = temp-, sibling;

Lo - silling = NULL;

heap-push-front (Lo);

Situm heap;

(4)

frame