TAMAL DATTA

Date:10/05/2020

1. **Design a function add_begin() using single linked list and print it.**

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_begin(abc **p);

void main()
{
   abc *hp=0;
   add_begin(&hp);
   puts("outut is:");
   print(hp);
}

void add_begin(abc **p)
{
    abc* temp;
    char op;
    do{
        temp=malloc(sizeof(abc));
        puts("enter  val=?");
        scanf(" %d",&temp->a);

        temp->n=*p;
        *p=temp;

        puts("add more nodes? y/n");
        scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}

void print(abc* p)
{
    while(p)
    {
        printf("%d\n",p->a);
        p=p->n;
    }
}
```

2. Design a function add_middle() using single linked list and print it.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_middle(abc **p);

void main()
{
   abc *hp=0;
   add_middle(&hp);
   puts("outut is:");
   print(hp);
}
void add_middle(abc **p)
{
    abc* temp;
    char op;
    do{
        temp=malloc(sizeof(abc));
        puts("enter val=?");
        scanf(" %d",&temp->a);

        if(*p==0 || (*p)->a>temp->a)
        {
            temp->n=*p;
            *p=temp;
        }
        else
        {
            abc* temp2=*p;
            while(temp2->n!=0 && temp2->n->a<temp->a)
                temp2=temp2->n;
            temp->n=temp2->n;
            temp2->n=temp;
        }

        puts("add more nodes? y/n");
        scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}
```

```
void print(abc* p)
{
    while(p)
    {
        printf("%d\n",p->a);
        p=p->n;
    }
}
```

3. Design a function add_end() using single linked list and print it.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_end(abc **p);

void main()
{
   abc *hp=0;
   add_end(&hp);
   puts("outut is:");
   print(hp);
}

void add_end(abc **p)
{
    abc* temp;
    char op;
    do{
        temp=malloc(sizeof(abc));
        puts("enter val=?");
        scanf(" %d",&temp->a);

        if(*p==0)
        {
            temp->n=*p;
            *p=temp;
        }

        else
        {
            abc* temp2=*p;
            while(temp2->n!=0)
               temp2=temp2->n;
            temp->n=temp2->n;
            temp2->n=temp;
         }

         puts("add more nodes? y/n");
         scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}

void print(abc* p)
{
    if(p)
    {
        printf("%d\n",p->a);
        print(p->n);
    }
}
```

4. **Design a function reverse_links() in single linked list and print it.**

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_end(abc **p);
int count(abc *p);
void rev_link(abc **p);

void main()
{
   abc *hp=0;
   add_end(&hp);
   puts("outut is:");
   print(hp);
   rev_link(&hp);
   puts("after reversing");
   print(hp);
}
```

```
void add_end(abc **p)
{
   abc* temp;
   char op;
   do{
      temp=malloc(sizeof(abc));
      puts("enter val=?");
      scanf(" %d",&temp->a);

      if(*p==0)
      {
         temp->n=*p;
         *p=temp;
      }

      else
      {
         abc* temp2=*p;
         while(temp2->n!=0)
            temp2=temp2->n;
         temp->n=temp2->n;
         temp2->n=temp;
      }


      puts("add more nodes? y/n");
      scanf(" %c",&op);
   }while(op=='y' || op=='Y');
}

int count(abc *p)
{
   int c=0;
   while(p)
   {
      c++;
      p=p->n;
   }
   return c;
}

void rev_link(abc **p)
{
 abc *a,*b,*c;
 a=*p;
 b=0;
 while(a)
    {
       c=b;
       b=a;
       a=a->n;
       b->n=c;
    }
*p=b;
}

void print(abc* p)
{
    if(p)
    {
       printf("%d\n",p->a);
       print(p->n);
    }
}
```

5. **Design a function reverse_data() in single linked list and print it.**

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_begin(abc **p);
void rev_data(abc **p);

void main()
{
   abc *hp=0;
   add_begin(&hp);
   puts("outut is:");
   print(hp);
   rev_data(&hp);
   puts("outut is:");
   print(hp);
}

void add_begin(abc **p)
```

```
{
   abc* temp;
   char op;
   do{
      temp=malloc(sizeof(abc));
      puts("enter  val=?");
      scanf(" %d",&temp->a);

      temp->n=*p;
      *p=temp;

      puts("add more nodes? y/n");
      scanf(" %c",&op);
   }while(op=='y' || op=='Y');
}

void print(abc* p)
{
   while(p)
   {
      printf("%d\n",p->a);
      p=p->n;
   }
  }

void rev_data(abc **p)
{
   int c=0;
   abc* temp=*p;
   while(temp)
      {
       c++;
       temp=temp->n;
      }
   int arr[c],i=0;
   temp=*p;
   while(temp)
   {
      arr[i++]=temp->a;
      temp=temp->n;
   }
   i--;
   temp=*p;
   while(i>=0)
   {
      temp->a=arr[i--];
```

```
        temp=temp->n;
    }
}
```

6. Design a function sort_data() in single linked list and print it.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_begin(abc **p);
void sort(abc **p);
void main()
{
   abc *hp=0;
   add_begin(&hp);
   puts("outut is:");
   print(hp);
   sort(&hp);
   puts("outut is:");
   print(hp);
}

void add_begin(abc **p)
{
    abc* temp;
    char op;
    do{
        temp=malloc(sizeof(abc));
        puts("enter  val=?");
        scanf(" %d",&temp->a);

        temp->n=*p;
        *p=temp;

        puts("add more nodes? y/n");
        scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}

void print(abc* p)
{
    while(p)
    {
       printf("%d\n",p->a);
       p=p->n;
    }
   }
void sort(abc **p)
{
    int c=0;
    abc* temp=*p;
    while(temp)
        {
         c++;
         temp=temp->n;
        }
    int arr[c],i=0;
    temp=*p;
    while(temp)
    {
       arr[i++]=temp->a;
       temp=temp->n;
    }
       for(int i=0;i<c-1;i++)
       for(int j=i+1;j<c;j++)
          if(arr[i]>arr[j])
             arr[i]=arr[j]-arr[i]+(arr[j]=arr[i]);

    i=0;
    temp=*p;
    while(i<c)
    {
       temp->a=arr[i++];
       temp=temp->n;
    }
    }
```

7. **Design a function merge_data() in single linked list and print it.**

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_begin(abc **p);
void merge(abc **p,abc *p1,abc *p2);
int len(abc *p);
void main()
{
  abc *hp1=0,*hp2=0,*hp=0;
  puts("enter SLL1");
  add_begin(&hp1);
  puts("SLL1 is:");
  print(hp1);

  puts("enter SLL2");
  add_begin(&hp2);
  puts("SLL2 is:");
  print(hp2);

  merge(&hp,hp1,hp2);
  puts("merged SLL is:");
  print(hp);
}

void add_begin(abc **p)
{
   abc* temp;
   char op;
   do{
      temp=malloc(sizeof(abc));
      puts("enter  val=?");
      scanf(" %d",&temp->a);

      temp->n=*p;
      *p=temp;

      puts("add more nodes? y/n");
      scanf(" %c",&op);
   }while(op=='y' || op=='Y');
}

void print(abc* p)
{
   while(p)
   {
      printf("%d\n",p->a);
      p=p->n;
   }
   }
```

```
int len(abc *p)
{
   int c=0;
   while(p)
   {
      c++;
      p=p->n;
   }
   return c;
}

void merge(abc **p,abc *p1,abc *p2)
{
   int c1=len(p1),c2=len(p2);
   printf("c1=%d,c=%d\n",c1,c2);
   int arr[c1+c2],i=0;
   abc *temp=p1;
   while(temp)
   {
      arr[i++]=temp->a;
      temp=temp->n;
   }
   temp=p2;
   while(temp)
   {
      arr[i++]=temp->a;
      temp=temp->n;
   }

   for(int i=0;i<c1+c2-1;i++)
      for(int j=i+1;j<c1+c2;j++)
         if(arr[i]<arr[j])
            arr[i]=arr[j]-arr[i]+(arr[j]=arr[i]);

   i=0;
   while(i<c2+c1)
   {
      temp=malloc(sizeof(abc));
      temp->a=arr[i++];

      temp->n=*p;
      *p=temp;
   }
}
```

8. Design a function search_data() in single linked list and print it.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

void print(abc* p);
void add_end(abc **p);
int search(abc *,int);
void main()
{
   abc *hp=0;
   puts("enter SLL");
   add_end(&hp);
   puts("SLL is:");
   print(hp);

   int v;
   puts("value to find in SLL?");
   scanf("%d",&v);

   int r=search(hp,v);
   r>=0?printf("%d is present at node %d",v,r+1):printf("%d is not present in SLL");
}

void add_end(abc **p)
{
    abc* temp;
    char op;
    do{
        temp=malloc(sizeof(abc));
        puts("enter  val=?");
        scanf(" %d",&temp->a);

        if(*p==0)
        {
            temp->n=*p;
            *p=temp;
        }
        else
        {
            abc *temp1=*p;
            while(temp1->n!=0)
                temp1=temp1->n;
            temp->n=temp1->n;
            temp1->n=temp;
        }

        puts("add more nodes? y/n");
        scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}

void print(abc* p)
{
    while(p)
    {
        printf("%d\n",p->a);
        p=p->n;
    }
   }

int search(abc *p,int v)
{
    int c=0;
    while(p)
        {
            if(p->a==v)
                return c;
            c++;
            p=p->n;
        }
    return -1;
}
```

9. Design a function save_file() in single linked list and store in file.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<fcntl.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;
```

```
void print(abc* p);
void add_end(abc **p);
void save(abc *p);

void main()
{
  abc *hp=0;
  add_end(&hp);
  puts("outut is:");
  print(hp);
  save(hp);
}

void save(abc *p)
{
   char s[20];
   FILE *fp=fopen("SLL","w");
   while(p)
   {
      fprintf(fp,"%d\n",(p->a));
      p=p->n;
   }
  /*
   puts("file name?");
   scanf("%s",s);
   int f=open(s,O_WRONLY|O_CREAT|O_RDONLY,0666);
   while(p)
   {
      write(f,&(p->a),sizeof(p->a));
      p=p->n;
   }
   */
}


void add_end(abc **p)
{
   abc* temp;
   char op;
   do{
      temp=malloc(sizeof(abc));
      puts("enter val=?");
      scanf(" %d",&temp->a);

      if(*p==0)
```

```
        {
           temp->n=*p;
           *p=temp;
        }

        else
        {
           abc* temp2=*p;
           while(temp2->n!=0)
              temp2=temp2->n;
           temp->n=temp2->n;
           temp2->n=temp;
        }


        puts("add more nodes? y/n");
        scanf(" %c",&op);
    }while(op=='y' || op=='Y');
}


void print(abc* p)
{
    while(p)
    {
        printf("%d\n",p->a);
        p=p->n;
    }
}
```

10. Design a function find_middle_node(),count_node() in single linked list and print it.

Soln:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct abc{
    int a;
    struct abc* n;
}abc;

typedef struct ret{
    abc* a;
    int c;
}ret;

void print(abc* p);
void add_end(abc **p);
int count(abc *p);
ret middle_node(abc* p);

void main()
{
  abc *hp=0;
  add_end(&hp);
  puts("outut is:");
  print(hp);
  printf("no. of nodes are %d\n",count(hp));
  if(count(hp)&1 !=0 )
      printf("index of middle node is %d, address is %p ,value is
%d\n",middle_node(hp).c,middle_node(hp).a,(middle_node(hp).a)->a);
  else
      puts("SLL has even number of nodes so there is no middle node");
}

void add_end(abc **p)
{
   abc* temp;
   char op;
   do{
      temp=malloc(sizeof(abc));
      puts("enter val=?");
      scanf(" %d",&temp->a);

      if(*p==0)
      {
         temp->n=*p;
         *p=temp;
      }

      else
      {
         abc* temp2=*p;
         while(temp2->n!=0)
            temp2=temp2->n;
         temp->n=temp2->n;
         temp2->n=temp;
      }


      puts("add more nodes? y/n");
      scanf(" %c",&op);
   }while(op=='y' || op=='Y');
}

int count(abc *p)
{
    int c=0;
    while(p)
    {
        c++;
        p=p->n;
    }
    return c;
}

ret middle_node(abc* p)
{
    ret temp;
    temp.c=count(p)/2;
    for(int i=0;i<temp.c;i++)
        p=p->n;
    temp.a=p;
    return temp;
}

void print(abc* p)
{
    if(p)
    {
        printf("%d\n",p->a);
        print(p->n);
    }
}
```