TAMAL DATTA

1.WAP to print first 5 prime numbers starting from 100.

```
#include<stdio.h>
int prime(int a)
   {
   int c=0;
   for(int i=1;i<=a/2;i++)
      if(! (a%i))
         c++;
   return (c==1)?1:0;
   }

void main()
{
   int count=0;
   for(int i=100;count!=5 ;i++)
      if(prime(i))
         {
            printf("%d\t",i);
            count++;
         }
}
```

2. WAP to reverse the bits of given number?

```
#include<stdio.h>
void main()
{
   int num;
   puts("enter the number");
   scanf("%d",&num);

   puts("before reversing");
   for(int i=0;i<=31;i++)
      printf("%d",num>>(31-i)&1);
   printf("\n");

   for(int i=0,j=31;i<j;i++,j--)
```

```
    if(num>>i&1 != num>>j&1)
        {
          num=num^(1<<i);
          num=num^(1<<j);
        }

  puts("after reversing");
  for(int i=31;i>=0;i--)
    printf("%d",num>>i &1);
  printf("\n");
}
```

3. WAP to toggle first 3 alternate bits from 7 th position for given integer using bitwise operator.

```
#include<stdio.h>
void main()
{
  int n;
  puts("enter the digit");
  scanf("%d",&n);

  puts("entered digit's binary");
  for(int i=31;i>=0;i--)
    printf("%d",n>>i&1);

  int c=0;
  for(int i=7;c!=3;i=i+2)
    {   c++;
        n=n^(1<<i);
    }
  puts("\nafter change");
  for(int i=31;i>=0;i--)
    printf("%d",n>>i&1);
}
```

4. WAP to print 3 rd perfect number.

```
#include<stdio.h>
```

```
void main()
{
   int c=0;
   for(int n=1;c!=3;n++)
      {
         int j=1,sum=0;
         while(j<n)
            {
              if(n%j==0)
                  sum+=j;
               j++;
            }
         if(sum==n)
           c++;
         if(c==3)
            printf("third perfect number is %d\n",n);

      }

}
```

5. WAP to separate all the even and odd elements from given array and store into two separate array

```
#include<stdio.h>

typedef struct num{
   int n;
   struct num* p;
}num;

void add(num **n,int num)
{
   struct num *temp=malloc(sizeof(num));
   temp->n=num;
   temp->p=*n;
   *n=temp;
}
```

```
void main()
{
num *even=0,*odd=0;
int n;
puts("enter number of elements in array");
scanf("%d",&n);

int arr[n];
puts("enter elements of array");
for(int i=0;i<n;i++)
scanf("%d",&arr[i]);

puts("entered array is");
for(int i=0;i<n;i++)
printf("%d\t",arr[i]);

for(int i=0;i<n;i++)
   if(arr[i] & 1==1)
      add(&odd,arr[i]);
   else
      add(&even,arr[i]);

puts("\neven array is");
while(even)
        {
         printf("%d\t",even->n);
         even=even->p;
        }

puts("\nodd array is");
while(odd)
        {
           printf("%d\t",odd->n);
           odd=odd->p;
        }
}
```

6. WAP to swap first and last word of given string if both the word having equal length.

```
#include<stdio.h>
#include<string.h>
void len_check(char*p)
{
   char *p1=p,*s=p;
   int c=0;
   while(*p1!=0x20)
      {
         c++;
         p1++;
      }
   while(*p1)
      {
         p1++;
      }
   p1--;
   while(*p1!=0x20)
      {
         c--;
         p1--;
      }
   if(c==0)
      {
         puts("sub-string size are same and replaceble");
         *p1++;
         while(*p!=0x20)
            {
               *p=*p1+*p-(*p1=*p);
                p++; p1++;
            }
         puts("new string is:");
         puts(s);
      }
   else{
      puts("sub-string size are not same hence not replaceble");
   }
```

```
}

void main()
{
   char s[50];
   puts("enter the string");
   gets(s);
   puts("written string is:");
   puts(s);

   len_check(s);
 }
```

7. WAP to impement strcmp() case insensitive ?

```
#include<stdio.h>
#include<string.h>
int mystrcmp(char *s1,char* s2)
{
   int i=0;
   while(*s1)
      {
        if(*s1==*s2  ||  *s1^(1<<5)==*s2)
           {
              s1++;s2++;
              continue;
           }
         else
            return 1;

      return 0;
      }
}

void main()
{
   char s1[50],s2[50];
   puts("enter string 1");
```

```
    gets(s1);

    puts("enter string 2");
    gets(s2);

    int n=mystrcmp(s1,s2);
    n?puts("unequal"):puts("equal");
}
```

8. WAP to check given string is palindrom or not?

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[50],s1[50];
    puts("enter string");
    gets(s);
    strcpy(s1,s);
    char t;
    int l=strlen(s);
    for(int i=0,j=l-1;i<j;i++,j--)
        {
            t=s[i];s[i]=s[j];s[j]=t;
        }
    if(strcmp(s,s1)==0)
        puts("palindrome");
    else
        puts("not palindrome");

}
```

9. WAP to implement atof () function?

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[10];
    puts("enter the deciman number");
    gets(s);
```

```
    int d=10,i;
    float n=0;

          while(s[i]!='.')
             {
                n=n*10+(s[i]-48);
                i++;
             }
          i++;
          while(s[i])
             {
                n=n+(float)(s[i]-48)/d;
                d=d*10;
                i++;
             }
       printf("%f",n);
}
```

10. WAP to sort the data in a string using selection sort?

```
#include<stdio.h>
#include<string.h>

void main()
{
   char s[20];
   puts("enter the string");
   gets(s);
   puts("entered string is");
   puts(s);
   char *p1,*p2;
   for(p1=s;p1<s+strlen(s)-1;p1++)
      for(p2=p1+1;p2<s+strlen(s);p2++)
            if(*p2<*p1 && *p1!=0x20 && *p2!=0x20)
               *p2=*p1+*p2-(*p1=*p2);

   puts("after sorting, new string is");
   puts(s);
}
```

11. WAP to find fibnocci series using recurrsion?

```
#include<stdio.h>
void fibo(int*);
int n;

void main()
{
 puts("length of series?");
 scanf("%d",&n);
 int arr[n];
 fibo(arr);
 for(int i=0;i<n;i++)
   printf("%d\t",arr[i]);
}

void fibo(int *arr)
{   static int i=2;
   arr[0]=1;
   arr[1]=1;
   arr[i]=arr[i-1]+arr[i-2];

   if(i!= n-1)
      {
         i++;
         fibo(arr);

      }
}
```

12. WAP to find sum of the digits of given number using recursion

```
#include<stdio.h>
int sod(int n)
{
   static int sum=0;
   if(n)
      {
         sum+=n%10;
         sod(n/10);
      }
```

```
    if(n==0)
        return sum;

}

void main()
{
    int n;
    puts("enter number");
    scanf("%d",&n);
    int sum=sod(n);
    printf("sum of digits of %d is %d",n,sum);
}
```

13. WAP to find strrev()using recursion

```
#include<stdio.h>
#include<string.h>

void rev(char* s1,char* s2)
{
if(s1<s2)
    {
        *s1=*s2+*s1-(*s2=*s1);
        rev(s1+1,s2-1);
    }

}

void main()
{
    char s[30];
    puts("enter string");
    gets(s);
    puts("entered string is:");
    puts(s);
    rev(s,s+strlen(s)-1);
    puts("reversed string is:");
    puts(s);
}
```

14. WAP to allocate dynamic memory for 5 strings, scan data and print data

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

void main()
{
   char **p;
   p=calloc(5,sizeof(char*));
   for(int i=0;i<5;i++)
      *(p+i)=calloc(1,20);

   puts("enter strings");
   for(int i=0;i<5;i++)
      gets(p+i);

   puts("entered strings are:");
   for(int i=0;i<5;i++)
      puts(p+i);

}
```

15. WAP to do multiplication operation of two matrix, allocate dynamic memory for matrix.

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
   int **m1,**m2,r1,r2,c1,c2,**out;
   puts("no of row in m1?");
   scanf("%d",&r1);
   puts("no of column in m1?");
   scanf("%d",&c1);
   puts("no of row in m2?");
   scanf("%d",&r2);
   puts("no of column in m2?");
   scanf("%d",&c2);
```

```
if(c1!=r2)
   {
      puts("multiplication not possible");
      return;
   }
m1=calloc(r1,sizeof(int*));
for(int i=0;i<c1;i++)
   m1[i]=calloc(c1,sizeof(int));

puts("enter elements of m1");
for(int i=0;i<r1;i++)
   for(int j=0;j<c1;j++)
      scanf("%d",&m1[i][j]);


m2=calloc(r2,sizeof(int*));
for(int i=0;i<r2;i++)
   m2[i]=calloc(c2,sizeof(int));

puts("enter elements of m2");
for(int i=0;i<r2;i++)
   for(int j=0;j<c2;j++)
      scanf("%d",&m2[i][j]);

out=calloc(r1,sizeof(int*));
for(int i=0;i<r1;i++)
      out[i]=calloc(c2,sizeof(int));

for(int i=0;i<r1;i++)
   for(int j=0;j<c2;j++)
      {
         for(int k=0;k<r2;k++)
            out[i][j]+=m1[i][k]*m2[k][j];

      }


puts("m1 is:");
for(int i=0;i<r1;i++)
   {
```

```
        for(int j=0;j<c1;j++)
            printf("%d\t",m1[i][j]);
         puts("");
        }

    puts("m2 is:");
    for(int i=0;i<r2;i++)
        {
        for(int j=0;j<c2;j++)
            printf("%d\t",m2[i][j]);
         puts("");
        }

    puts("matrix multiplication result:");
    for(int i=0;i<r1;i++)
        {
        for(int j=0;j<c2;j++)
            printf("%d\t",out[i][j]);
         puts("");
        }

    for(int i=0;i<r1;i++)
            free(out[i]);
    free(out);

    for(int i=0;i<r1;i++)
            free(m1[i]);
    free(m1);
    for(int i=0;i<r1;i++)
            free(m2[i]);
    free(m2);

}
```

16. WAP to find second smallest and biggest element in an array

```
#include<stdio.h>
void main()
```

```
{
   int n;
   puts("enter no. of elements");
   scanf("%d",&n);
   puts("enter the array elements");
   int arr[n];
   for(int i=0;i<n;i++)
      scanf("%d",&arr[i]);

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

   printf("largest number=%d\n2nd smallest number=%d\n",arr[n-1],arr[1]);
}
```

17. WAP to copy source file to n no.of destination files(Command line argumnets)

```
#include<stdio.h>
#include<stdlib.h>

void main(int argc,char **argv)
{

   if(argc<3)
      {
         puts("usage:./a.out s_file d1_file d2_file .....");
         return;
      }


   FILE *fp=fopen(argv[1],"r");
   int i=2;
   char ch;
   while(i<argc)
      {
         FILE *f=fopen(argv[i],"w");
         while((ch=fgetc(fp))!=EOF)
            fputc(ch,f);
```

```
        rewind(fp);
        fclose(f);
        i++;
    }
fclose(fp);
}
```

18. Convert data in a file to opposite case in same file

```
#include<stdio.h>
#include<stdlib.h>

void main()
{
  if(argc!=2)
    {
      puts("usage:./a.out file_name");
      return;
    }

  FILE *fp=fopen(argv[1],"r+");
  if(fp==0)
  {
    perror("fopen");
     return;
  }
  char ch;
  while((ch=fgetc(fp))!=EOF)
          {
          if(ch>='A' && ch<='Z'   ||ch>='a' && ch<='z')
            {
            fseek(fp,-1,SEEK_CUR);
            fputc(ch^(1<<5),fp);
            }
          }
  fclose(fp);
}
```

19. Write a program to implement single linked list with the following functions. A) add_middle() B) delete() 3) display() 4) save()

```
#include<stdio.h>
#include<stdlib.h>

typedef struct link
{
   int num;
   struct link *next;
}sll;

void add(sll **p);
void del(sll **p);
void disp(sll *p);
void save(sll* p);

void main()
{
   sll *hp=0;
   char op;
   while(1)
   {
      L:
      puts("\nwant to add a link? Y/N/y/n");
      scanf(" %c",&op);
      if(op=='Y'  || op=='y' )
         {
            add(&hp);
            goto L;
         }

      puts("want to display the chang in linked list? Y/N/y/n");
      scanf(" %c",&op);
      if(op=='Y'  || op=='y' )
         disp(hp);

      L1:
      puts("\nwant to delete a link? Y/N/y/n");
      scanf(" %c",&op);
      if(op=='Y'  || op=='y' )
         {
            del(&hp);
```

```
            goto L1;
        }

    puts("want to display the chang in linked list? Y/N/y/n");
    scanf(" %c",&op);
    if(op=='Y'  || op=='y' )
       disp(hp);


    puts("\nwant to save a linked list? Y/N/y/n");
    scanf(" %c",&op);
    if(op=='Y'  || op=='y' )
       {
          save(hp);
       }


    puts("\nforgetting anything? Y/N/y/n");
    scanf(" %c",&op);
    if(op=='N'  || op=='n' )
       break;
  }

  puts("**********************closed**************************");
}


void add(sll **p)
{
sll *temp=malloc(sizeof(sll)),*q=*p;
puts("enter the number");
scanf("%d",&(temp->num));

if(*p==0 || (*p)->num>temp->num)
  {
    temp->next=*p;
    *p=temp;
  }
else
  {
```

```
      while(1)
         {
            if((q)->next==0    || (q)->next->num>temp->num)
               {
                  temp->next=q->next;
                  q->next=temp;
                  break;
               }
            q=(q)->next;
         }
   }
}


void disp(sll *p)
{
   if(p)
      {
         printf("%d\t",p->num);
         disp(p->next);
      }
}

void del(sll **p)
{
   sll *q=*p,*q1=*p;
   int n;
   puts("enter the number to delete");
   scanf("%d",&n);

   while(1)
      {
         if(q==0)
            {
               printf("number is not present in linked list ");
               break;
            }
         else if(q->num==n)
            {
               if(q==*p)
```

```
                {
                   *p=q->next;
                }
                else
                {
                while(q1->next!=q)
                   q1=q1->next;
                q1->next=q1->next->next;
                }
             break;
             }


       q=q->next;

       }
}



void save(sll* p)
{
   char s[20];
   puts("enter the file name");
   scanf("%s",s);
   FILE *fp=fopen(s,"w");
   while(p)
       {
       fprintf(fp,"%d ",(p->num));
       p=p->next;
       }
   fclose(fp);

}
```

20. Write a program to implement stack using Arrays.

```
#include<stdio.h>
#include<stdlib.h>
```

```
int count=0;

void push(int** p,int num)
{
int *temp=malloc((count+1)*sizeof(int));
int i;
for(i=0;i<count;i++)
 temp[i]=(*p)[i];
temp[count]=num;
*p=realloc(*p,(count+1)*sizeof(int));
for(int i=0;i<=count;i++)
   (*p)[i]=temp[i];
free(temp);
}


void pop(int** p)
{
int *temp=malloc((count-1)*sizeof(int));
for(int i=0;i<count-1;i++)
   temp[i]=(*p)[i];
*p=realloc(*p,(count-1)*sizeof(int));
for(int i=0;i<count-1;i++)
   (*p)[i]=temp[i];
free(temp);
}




void main()
{
   int *p,num;
   char op;
   while(1)
       {
       puts("\nwant to push a value to stack?");
       scanf(" %c",&op);
       if(op=='Y' || op=='y')
```

```
        {
        L:  puts("enter the number");
            scanf("%d",&num);
            push(&p,num);
            count++;
            puts("want to push one more value to stack?");
            scanf(" %c",&op);
                if(op=='Y' || op=='y')
                    goto L;

        if(sizeof(p))
        {
            puts("New stack is");
            for(int i=0;i<count;i++)
                printf("%d\t",p[i]);
        }
        else
            puts("stack is empty");

        }

    puts("\nwant to pop a value from stack?");
    scanf(" %c",&op);
    if(op=='Y' || op=='y')
        {
        L1: pop(&p);
            count--;
            puts("want to pop one more value from stack?");
            scanf(" %c",&op);
                if(op=='Y' || op=='y')
                    goto L1;

        puts("New stack is");
        for(int i=0;i<count;i++)
            printf("%d\t",p[i]);
        }


    }
}
```