ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 4, April 2018

Computerisation of Vedic Mathematics

[1] K.Palani, [2] T.Narmatha
[1] Associate Professor of Mathematics, [2] M.Phil Scholar
[1][2] A.P.C Mahalaxmi College For Women Thoothukudi, Tamilnadu India.

Abstract: **- To carry out tedious and cumbersome mathematical operations, simple formulae are in the book Vedic Mathematics by Bharathi Krishna Thirathaji Maharaj. In this article, we made an attempt to write c++ programs for multiplication of two "2 digit numbers", multiplication of any number by 11 and by 9 and division of any polynomial of degree n by any linear polynomial. as explained in Vedic Mathematics.**

Keywords: Cleanliness, promoting cleanliness awareness.

I. INTRODUCTION

Vedic Mathematics widens the perspective of viewing at and knowing mathematics. It simplifies and reveals mathematics in a different light. It bypasses the customary, complicated steps in solving problems and shows quick methods in solving the same. In short, Vedic mathematics enhances the skill to approach and solve any problem to get right solution quickly. Hoping at increasing the speed of computers using these Vedic mathematics, we start with the c++ programs for the basic algebraic operation multiplication. Further, we added the programs for division of any polynomial of degree n by linear polynomial.

II MULTIPLICATION BY n-DIGIT MULTIPLIER WITH EACH DIGIT EQUAL TO 9:

2.1 ALGORITHM:

Assume that the number of digits in the multiplier is equivalent to the number of digits in the multiplicand.
Step 1: Subtract 1 from multiplicand. Let the answer be x.
Step 2: Subtract the x from multiplier. Let it be y.
Step 3: The answer is x | y.

2.2 ILLUSTRATION:

To find 234567 * 999999
Step 1: 234567 - 1 = 234566
Step 2: 999999-234566 =765433
Step 3: 234567*999999=234566 | 765433
=234566765433

2.3 PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main ( )
longinta,b,c,d,e,f,count=0,count 1=0;
clrscr ( );
printf("Enter a number");
scanf("% 1d",&a);
printf("Enter the second number");
scanf("%1d",&b);
c=a;
d=b;
while(d>0)
{
d=d/10;
count 1=count 1+1;
}
printf("count%1dcount1%1d",count,count1");
if(count!=count1)
{
printf("please check the input");
}
else
{
e=a-1;
f=b-e;
printf("ans=1%d%1d",e,f);
}
getch( );
}
```

OUTPUT:

```
Enter a number
234567
Enter the second number 999999
Count=6 count1=6 ans=234566765433
Enter a number
1234567876
Enter a second number 9999999999
Count=10 count1=10
ans =1234567875912915772
```

2.4 REMARK

It is verified that the above program on execution produces the exact solution.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 4, April 2018

III THE PRODUCT OF ANY 'n' DIGIT NUMBERS BY 11:

3.1 ALGORITHM:

Step 1: Take the n digit number as x.
Step 2: The one's digit of the answer 'y' in the one's digit of x.
Step 3: Tens digit of y = Tens digit of x + Ones digit of x.
Step 4: Hundreds digit of y = Hundreds digit of x + Tens digit of x.
Step 5: Repeat the above procedure for each other of y.

3.2 ILLUSTRATION:

To find 1421 * 11
Step 1: x = 1421
Step 2: y = 1
Step 3: y = 31
Step 4: y = 15631
Step 5: 1421 * 11 = 15631(y).

3.3 PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
intstl,i,j;
char a[20];
intaa[20];
clrscr( );
printf("Enter a number");
scanf("%s",&a);
stl=strlen(a);
aa[0]=a[0]-48;
for(i=0;i<stl-1;i++)
{
aa[i+1]=((a[i]-48)+9a[i+1]-48));
}
for(i=0;i<s+1;i++)
{
if(aa[i]>9)
{
aa[i]=aa[i]-10;
aa[i-1]++;
}}
for(i=0;i<=stl-1;i++)
{
printf("%d",aa[1]);
}
printf("%d",a[stl-1]-48);
getch( );
}
```

OUTPUT:

```
Enter a number 1421
15631
Enter a number 852741964783214
93710161612615354
```

3.4 REMARK:

It is verified that the above program on execution produces the exact solution.

IV THE PRODUCT OF ANY TWO "TWO DIGIT NUMBERS"

4.1 ALGORITHM:

Step 1: Multiply the left-hand most digit of the multiplicand and multiplier vertically, get their product and set it down as the left-hand most part of the answer.
Step 2: Then multiply the multiplicand and multiplier cross-wise, add the two, get their sum and set it down as the middle part of the answer.
Step 3: Multiply the right hand side-most digit of multiplicand and multiplier vertically, get their product and set it down as the last right hand most partof the answer.
Step 4: Therefore the answer is x | y | z. If y or z has more than one digit then add the left most digit with the previous one. That is if z has two digits a | b then assign z = b and add 'a' with y.

4.2 ILLUSTRATION:

To find 23 * 15

Step 1: 2 * 1 = 2

```
   23
   15
 -----
    2
```

Step 2: 2 * 5 +3 * 1 = 10 + 3 = 13

```
   23
   15
 ------
   23
    1
```

Step 3: 5 * 3 = 15

```
  23
     15
  -----
   235
    11
  -----
   345
```

4.3 PROGRAM:

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

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 4, April 2018

```
void main( )
{
char a[2],b[2],c[2];
intaa[2],bb[2],f,d,e,g,h,i;
clrscr( );
printf("Enter the numbers");
cets(c);
gets(b);
printf("a=%s",a);
aa[0]=c[0]-48;
aa[1]=c[1]-48;
bb[0]=b[0]-48;
bb[1]=b[1]-48;
g=(aa[1]*bb[1]);
d=aa[0]*bb[0];
printf("%d%d%d%d",aa[0],aa[1],bb[0],bb[1]);
f=((aa[0]*bb[1])+(bb[0]*aa[1]));
if(f>9)
{
h=f%10;
i=f\10;
printf("ans=%d%d%d",d+I,h,g);
}
Else
{
Printf("ans=%d%d%d",d,f,g);
}
getch( );
}
```

OUTPUT:
Enter a number=22
24
a=2224 ans =528

4.4 REMARK:

It is verified that the above program on execution produces the exact solution.

V VEDIC METHOD OF DIVISION OF ANY POLYNOMIAL OF DEGREE N BY LINEAR POLYNOMIAL.

5.1 METHOD EXPLAINED IN VEDIC MATHS:

Assume that the dividend $a_0x^n + a_1x^{n-1}+a_2x^{n-2}+\ldots\ldots+a_n$ and the divisor is x-a.

Let $b_0x^{n-1}+b_1x^{n-2}+\ldots\ldots+b_{n-1}$ be the quotient and b_n be the remainder.

Now , to find b_i for all i = 0 to n.

Let $b_0 = a_0$ and a' = -a

$b_1 = a_0a'+a_1$

$b_2 = a_1a'+a_2$

.

$b_n= a_na'+a_n$

Now, the quotient = $b_0x^{n-1}+b_1x^{n-2}+\ldots\ldots+b_{n-1}$ and remainder is b_n.

5.2 ALGORITHM:

Step: 1

Input the degree of dividend n.

Step: 2

For i = 0 to n, Input a_i

Step: 3

Put $b_0=a_0$

Step: 4

For i = 1 to n

$b_i= a_ia'+a_i + 1$

Step: 5

Print dividend $=b_0x^{n-1}+b_1x^{n-2}+\ldots\ldots+b_{n-1}$ Remainder = b_n.

5.3 PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n[100],c[100],i,a,b,g,e,d,f;
clrscr();
printf("enter the highest degree");
scanf("%d",&n[0]);
f=n[0];
for(i=0;i<=n[0];i++)
{
printf("\n_x^%d",f);
scanf("%d",&c[i]);
n[i]=f;
f--;
}
for(i=0;i<=n[0];i++)
{
printf("%+dx^%d",c[i],n[i]);
}
printf("enter the divisor x");
scanf("%d",&a);
printf("divisor=x%+d",a);
printf("\n quotient=%+dx^%d",c[0],n[0]-1);
d=c[0];
for(i=0;i<n[0]-1;i++)
{
b=-(d*a);
d=b+c[i+1];
printf("%+dx^%d",d,n[i]-2);
}
g=-(d*a)+c[n[0]];
printf("reminder=%+d",g);
getch();
}
```

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 3, Issue 4, April 2018

5.4 OUTPUT:

12 x^2 – 8x – 32 is divided by x-2

Quotient = 12 x + 16

Remainder = 0

5.5 REMARK:

It is verified that the above program on execution produces the exact solution.

VI CONCLUSION

Vedic mathematics when practiced yields fastest calculating talent to our mind. So, the computers which is hoped for doing computation much much faster than the human brain may work faster than the present speed if vedic mathematics programs are used in software .This is the initiation of that task.

REFERENCE:

[1].E.BalaGurusamy-Fundamentals of Computing and Programming-Copyright 2010 by the Tata McGraw Hill Education Private Limited-First Reprint.

[2].Bharati Krisna Tirthaji Maharaj-Vedic Mathematics - 17thReprint : Delhi 2015.

[3].Kenneth R.Williams-Vedic Mathematics-Teacher'sManual[Intermediate level]-First Indian Edition:Delhi 2005.