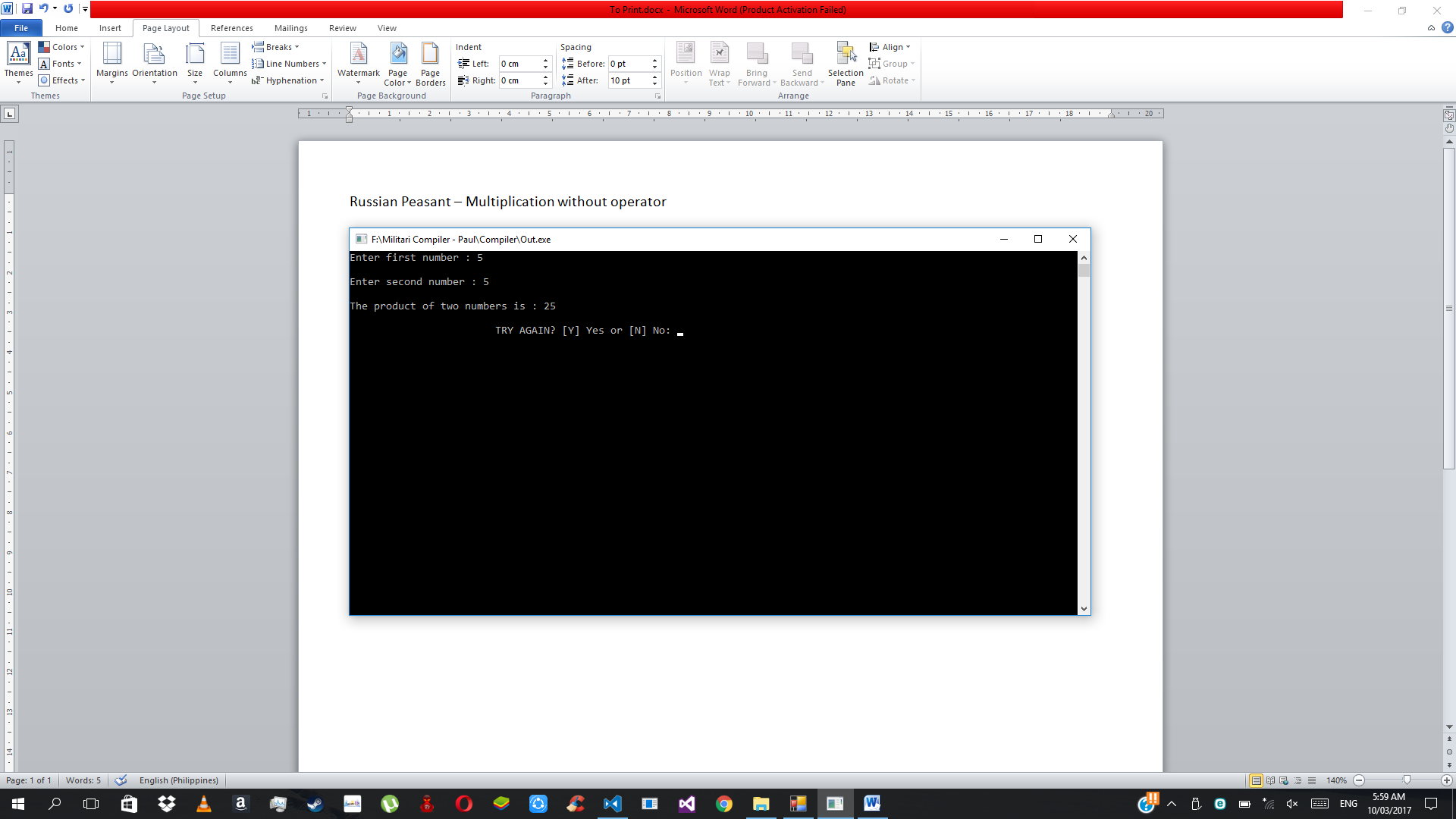
XXVI. SAMPLE PROGRAMS

Russian Peasant – Multiplication without operator

Sample Input – Output:



Source Code:

PrimaryMission() {

unit a,b,p=0,mod1,mod2;

unit choice;

go {

commence;

post("Enter first number : ");

capture(#a);

post("\nEnter second number : ");

capture(#b);

mod1 = a%2;

inorder(mod1!=0) {

p=p+b;

}

phase(a!=1) {

a=a/2;

b=b\*2;

mod2 = a%2;

inorder(mod2!=0) {

p=p+b;

}

}

post("\nThe product of two numbers is : " + p);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

a=0;

b=0;

p=0;

mod1=0;

mod2=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

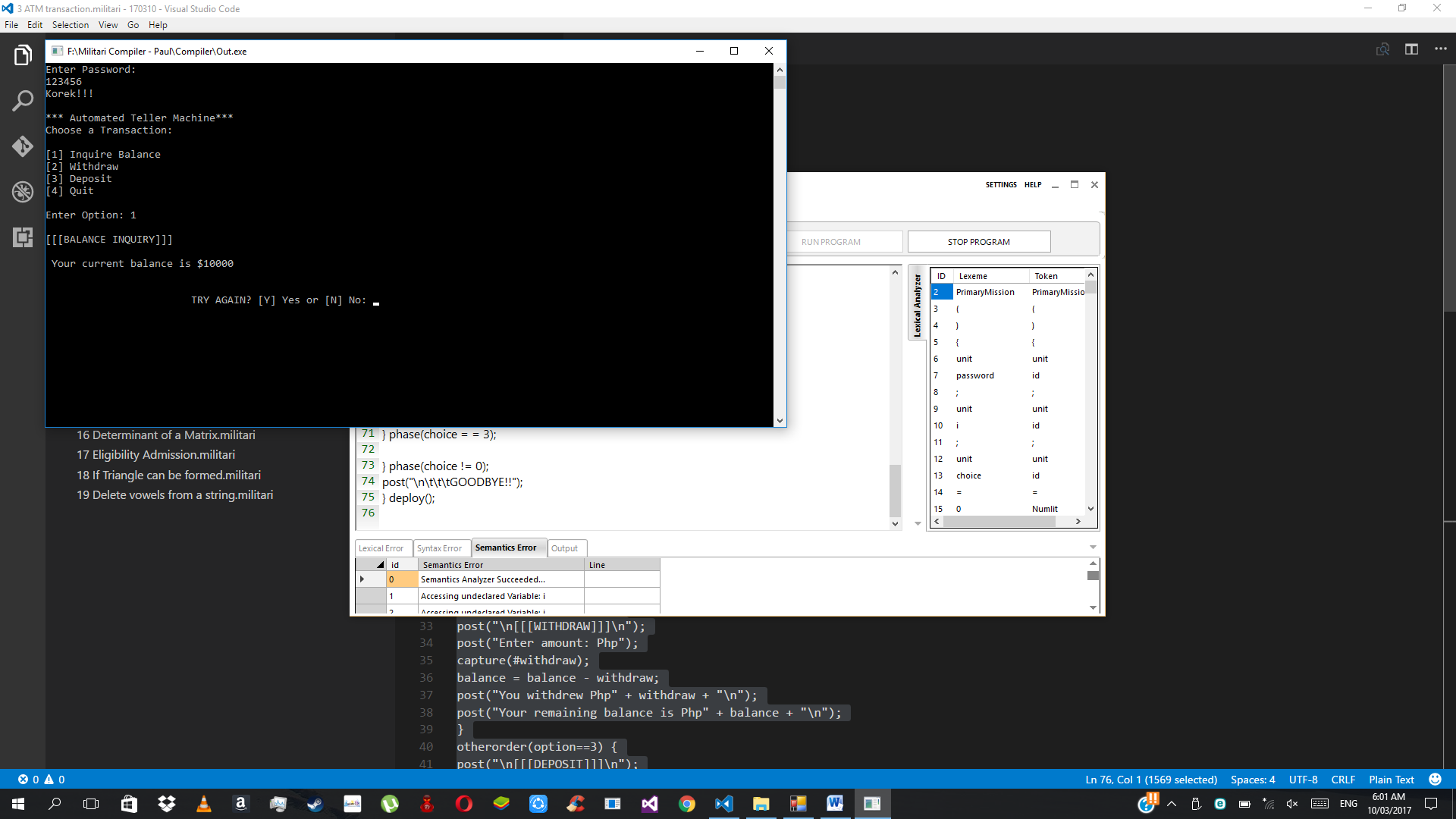
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

ATM Transaction

Sample Input – Output:



Source Code:

PrimaryMission() {

unit password;

unit i;

unit choice=0;

go {

commence;

inquire(i=0;i<3;i++) {

post("Enter Password:\n");

capture(#password);

inorder(password==123456) {

post("Korek!!!\n");

digit balance = 10000;

digit withdraw, deposit;

unit option;

post("\n");

post("\*\*\* Automated Teller Machine\*\*\*\n");

post("Choose a Transaction:\n");

post("\n");

post("[1] Inquire Balance \n");

post("[2] Withdraw \n");

post("[3] Deposit \n");

post("[4] Quit \n");

inorder(option==1) {

post("\n[[[BALANCE INQUIRY]]]\n");

post("\n Your current balance is $" + balance + "\n");

abort();

}

otherorder(option==2) {

post("\n[[[WITHDRAW]]]\n");

post("Enter amount: Php");

capture(#withdraw);

balance = balance - withdraw;

post("You withdrew Php" + withdraw + "\n");

post("Your remaining balance is Php" + balance + "\n");

}

otherorder(option==3) {

post("\n[[[DEPOSIT]]]\n");

post("Enter amount: Php");

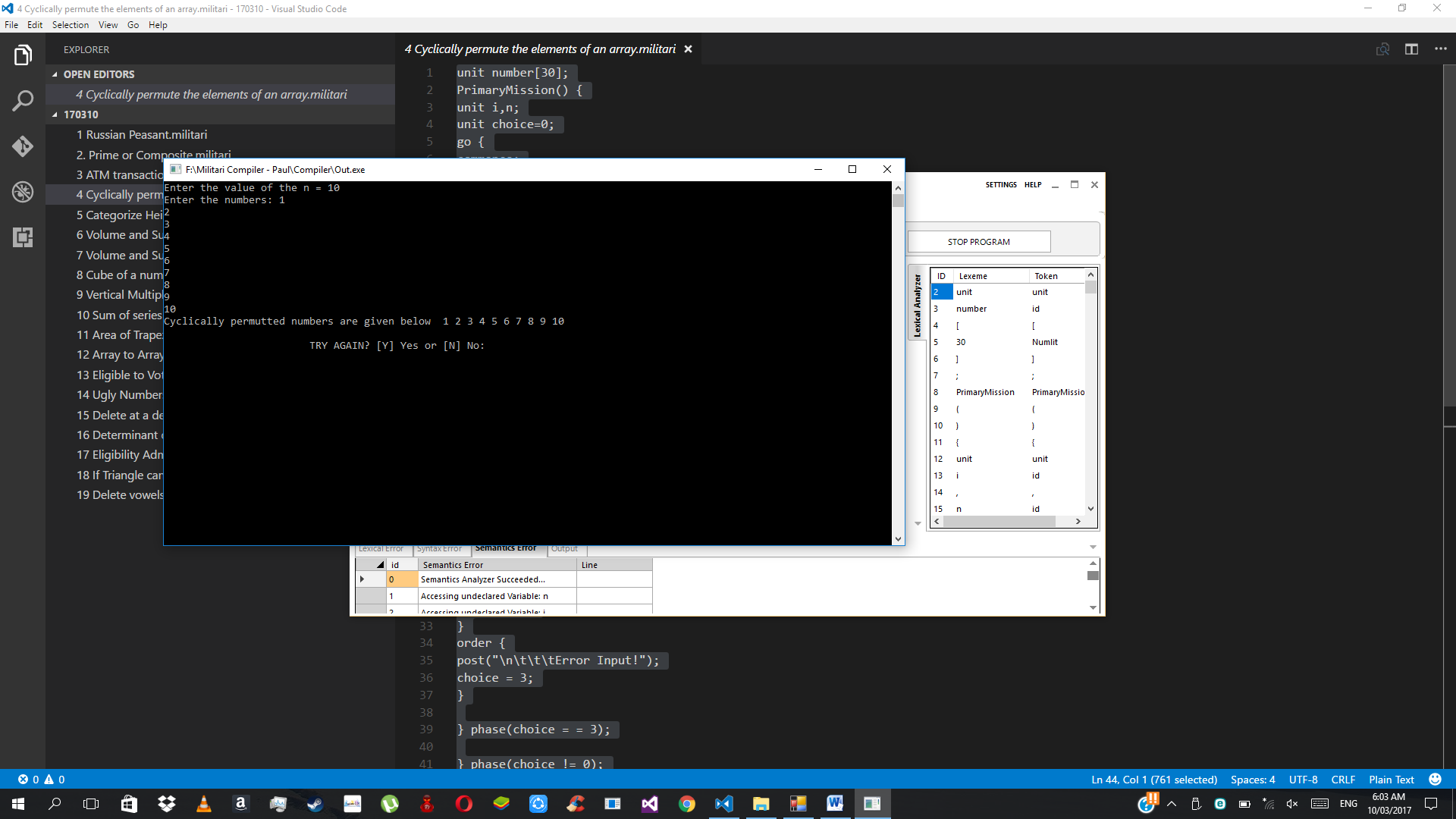
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Cyclically Permute the Elements of an Array

Sample Input – Output:



Source Code:

unit number[30];

PrimaryMission() {

unit i,n;

unit choice=0;

go {

commence;

post("Enter the value of the n = ");

capture(#n);

post("Enter the numbers: ");

inquire(i=0;i<n;i++) {

capture(#number[i]);

}

number[n] = number[0];

inquire(i=0;i<n;i++) {

number[n] = number[i+1];

}

post("Cyclically permutted numbers are given below ");

inquire(i=0;i<n;i++) {

post(" " + number[i]);

}

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

i=0;

n=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

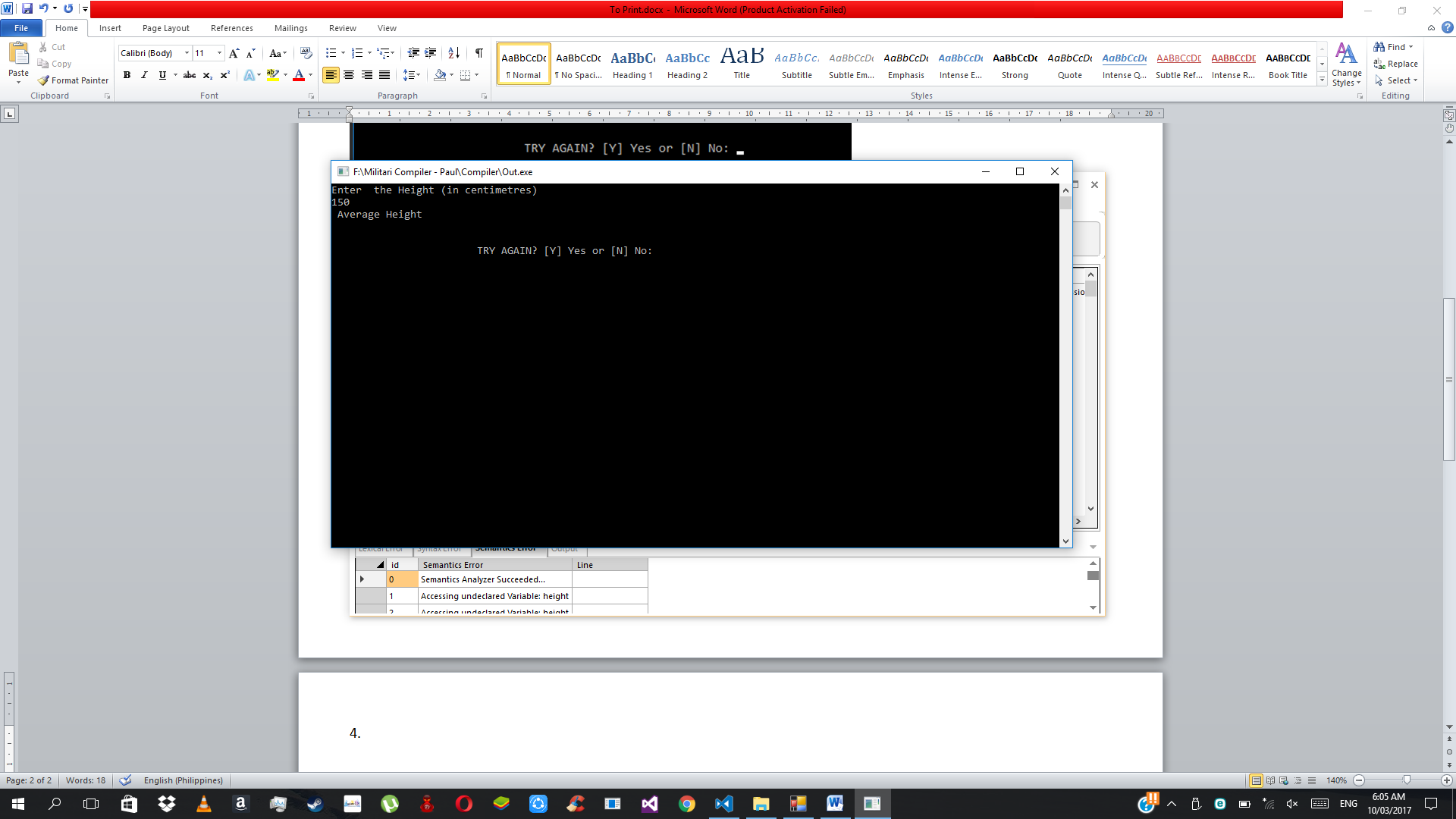
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Categorize Height

Sample Input – Output:



Source Code:

PrimaryMission() {

digit height;

unit choice=0;

go {

commence;

post("Enter the Height (in centimetres) \n");

capture(#height);

inorder(height < 150.0) {

post("Dwarf \n");

}

otherorder((height >= 150.0) & (height <= 165.0)) {

post(" Average Height \n");

}

otherorder((height >= 165.0) & (height <= 195.0)) {

post("Taller \n");

}

order {

post("Abnormal height \n");

}

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

height=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

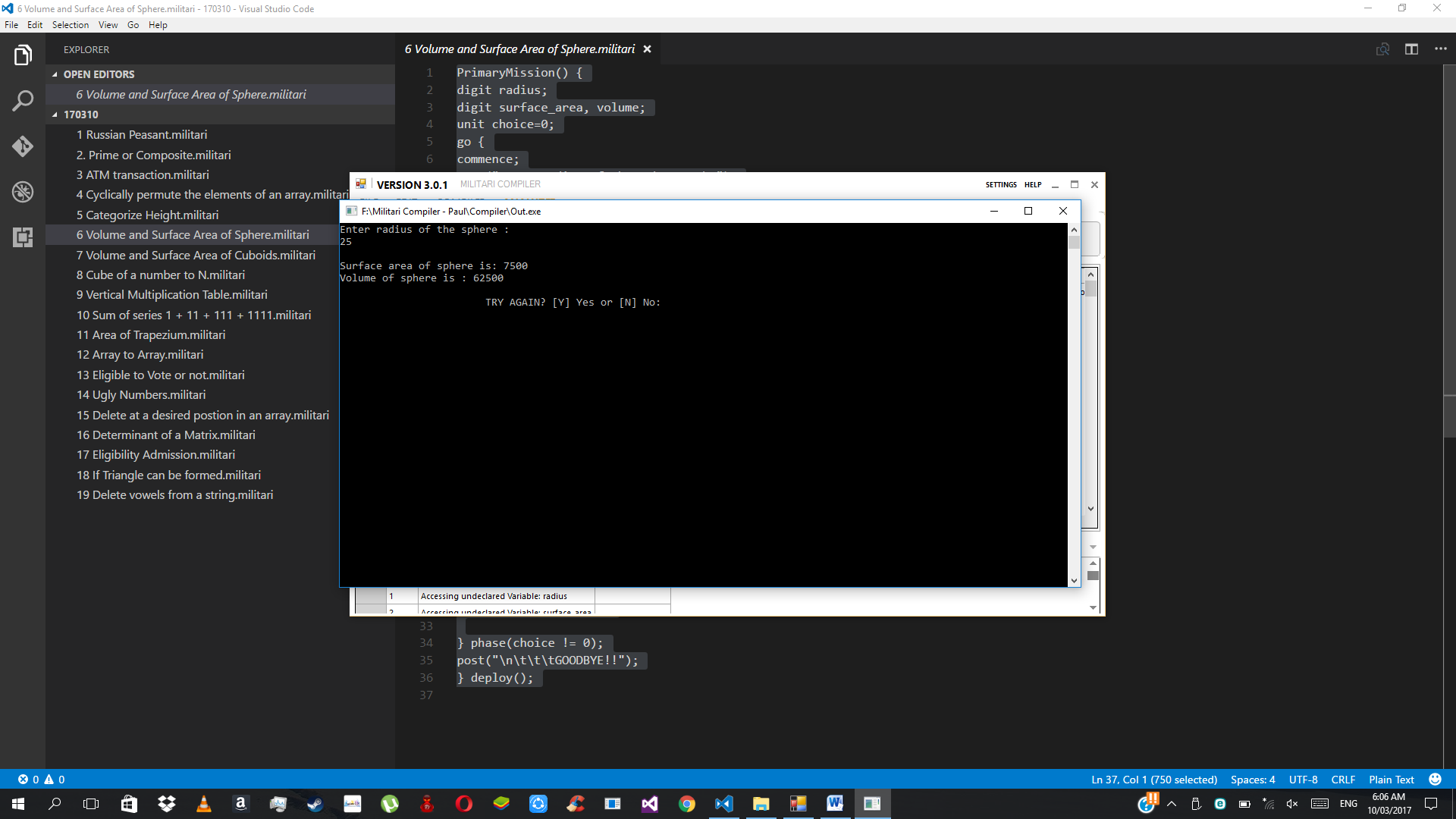
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Volume and Surface Area of a Sphere

Sample Input – Output:



Source Code:

PrimaryMission() {

digit radius;

digit surface\_area, volume;

unit choice=0;

go {

commence;

post("Enter radius of the sphere : \n");

capture(#radius);

surface\_area = 4 \* (22/7) \* radius \* radius;

volume = (4.0/3) \* (22/7) \* radius \* radius \* radius;

post("\nSurface area of sphere is: " + surface\_area);

post("\nVolume of sphere is : " + volume);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

radius=0;

surface\_area=0;

volume=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

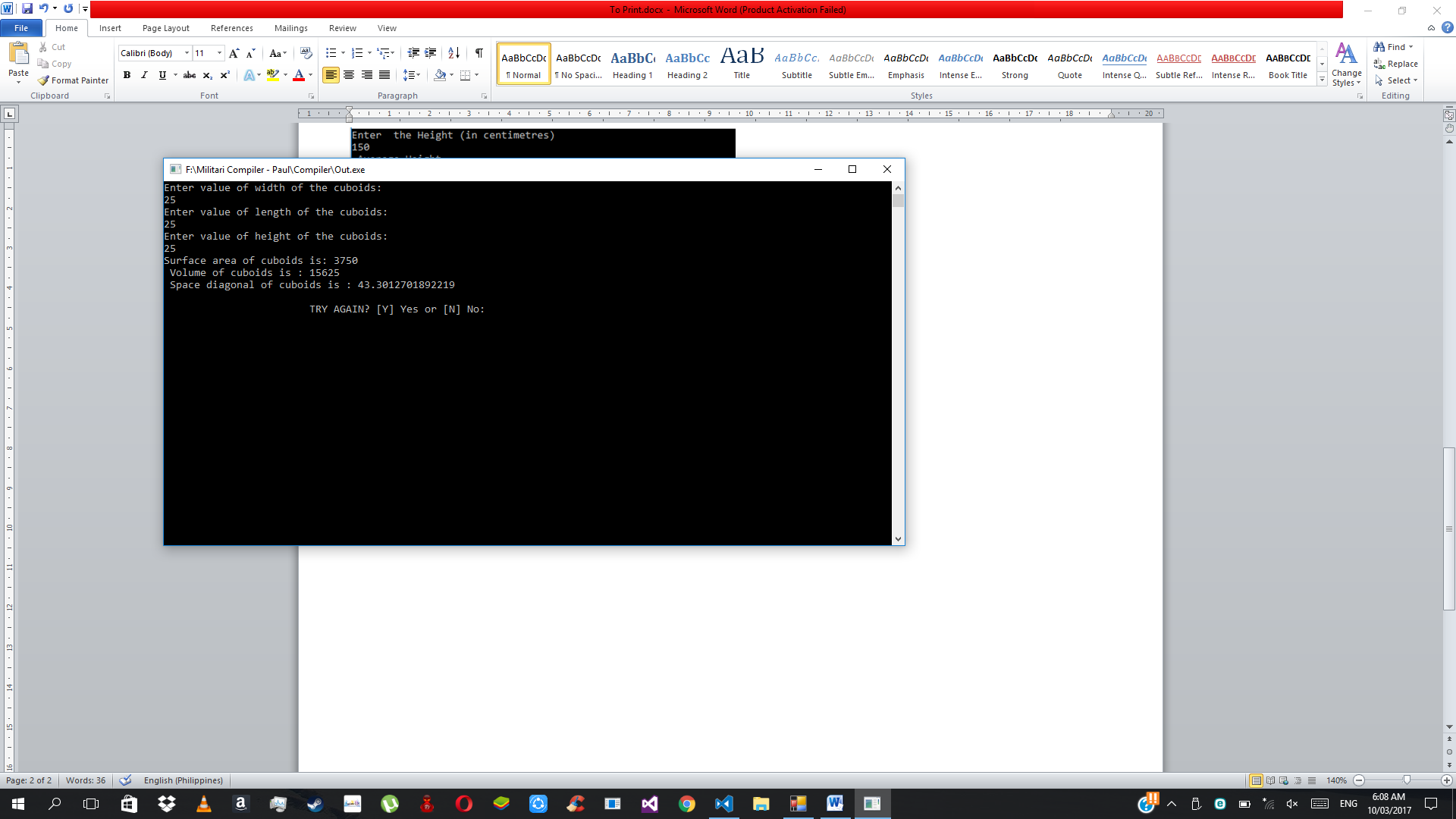
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Volume and Surface Area of a Cuboid

Sample Input – Output:



Source Code:

PrimaryMission() {

digit width, length, height,temp,temp2;

digit surfacearea, volume, space\_diagonal;

unit choice=0;

go {

commence;

post("Enter value of width of the cuboids:\n");

capture(#width);

post("Enter value of length of the cuboids:\n");

capture(#length);

post("Enter value of height of the cuboids:\n");

capture(#height);

temp = width \* length + length \* height + height \* width;

surfacearea = 2 \* temp;

volume = width \* length \* height;

temp2 = width \* width + length \* length + height \* height;

space\_diagonal = sqrt(temp2);

post("Surface area of cuboids is: " + surfacearea);

post("\n Volume of cuboids is : " + volume);

post("\n Space diagonal of cuboids is : " + space\_diagonal);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

width=0; length=0;

height=0; temp=0;

temp2=0; surfacearea=0;

volume=0;

space\_diagonal=0;

choice = 1;

}otherorder((ch = = "N") || (ch = = "n")) { choice = 0; }

order { post("\n\t\t\tError Input!"); choice = 3;

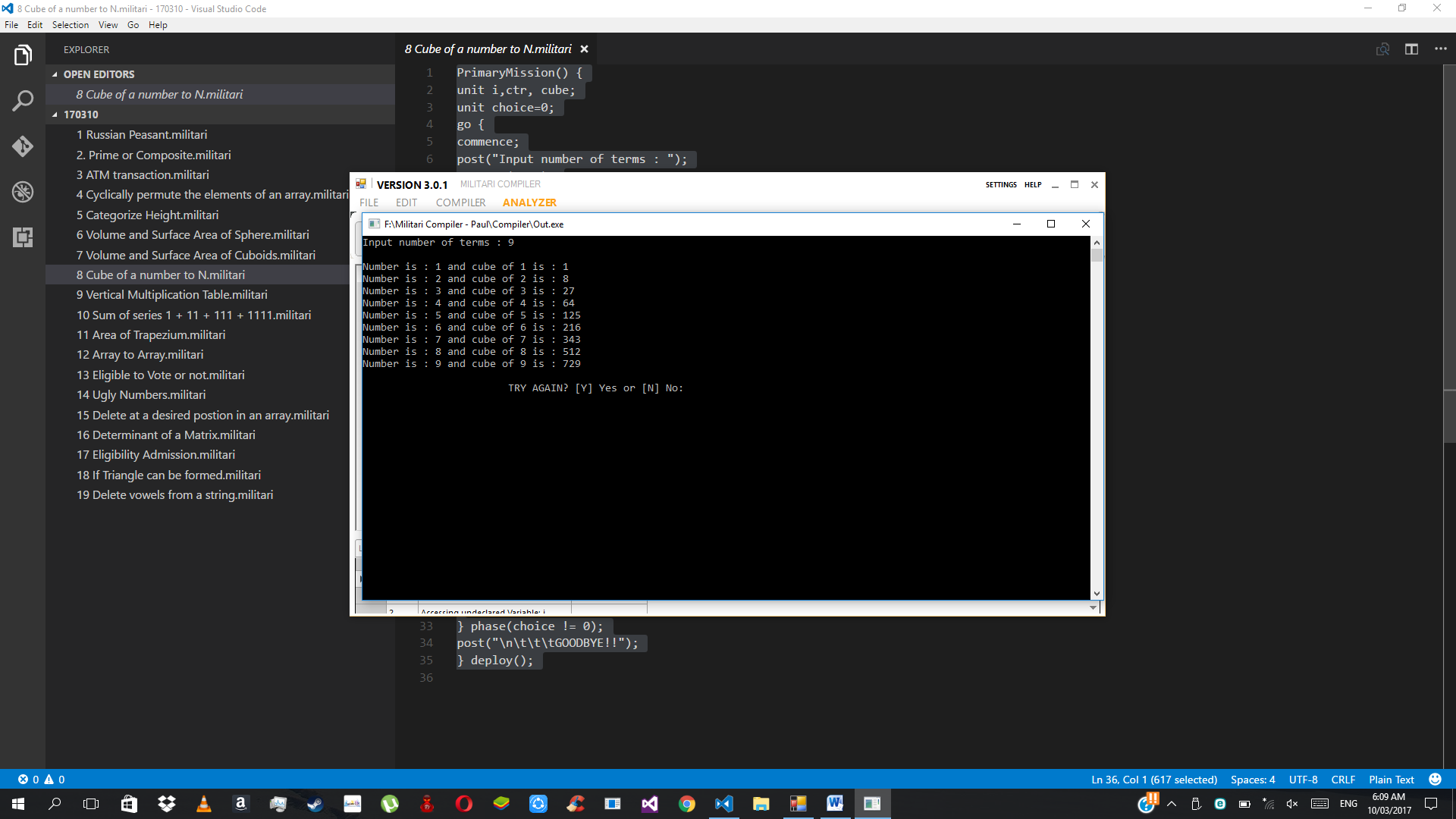
} } phase(choice = = 3);

} phase(choice != 0); post("\n\t\t\tGOODBYE!!");

} deploy();

Cube of a Number to N

Sample Input – Output:



Source Code:

PrimaryMission() {

unit i,ctr, cube;

unit choice=0;

go {

commence;

post("Input number of terms : ");

capture(#ctr);

inquire(i=1;i<=ctr;i++) {

cube = i\*i\*i;

post("\nNumber is : " + i + " and cube of " + i + " is : " + cube);

}

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

i=0;

ctr=0;

cube=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

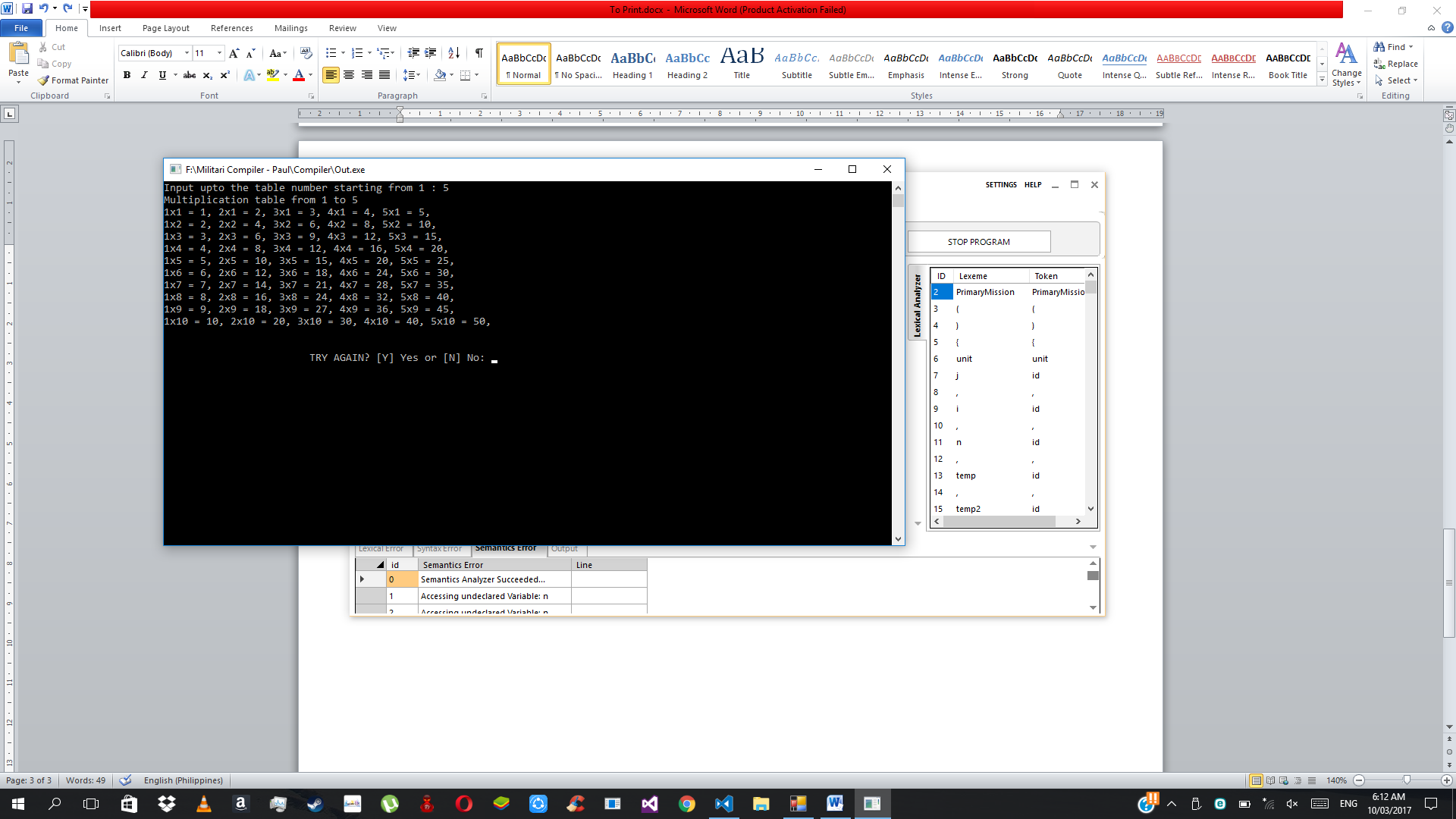
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Vertical Multiplication Table

Sample Input – Output:



Source Code:

PrimaryMission() {

unit j,i,n,temp,temp2,temp3;

unit choice=0;

go {

commence;

post("Input upto the table number starting from 1 : ");

capture(#n);

post("Multiplication table from 1 to " + n + " \n");

inquire(i=1;i<=10;i++) {

inquire(j=1;j<=n;j++) {

temp3=n-1;

inorder(j<=temp3) {

temp=i\*j;

post(j + "x" + i + " = " + temp + ", "); }

order { temp2=i\*j;

post(j + "x" + i + " = " + temp2 + ", "); } }

post("\n"); }

go { company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

j=0;

i=0;

n=0;

temp=0;

temp2=0;

temp3=0;

choice = 1;

} otherorder((ch = = "N") || (ch = = "n")) { choice = 0;

} order {

post("\n\t\t\tError Input!");

choice = 3; }

} phase(choice = = 3);

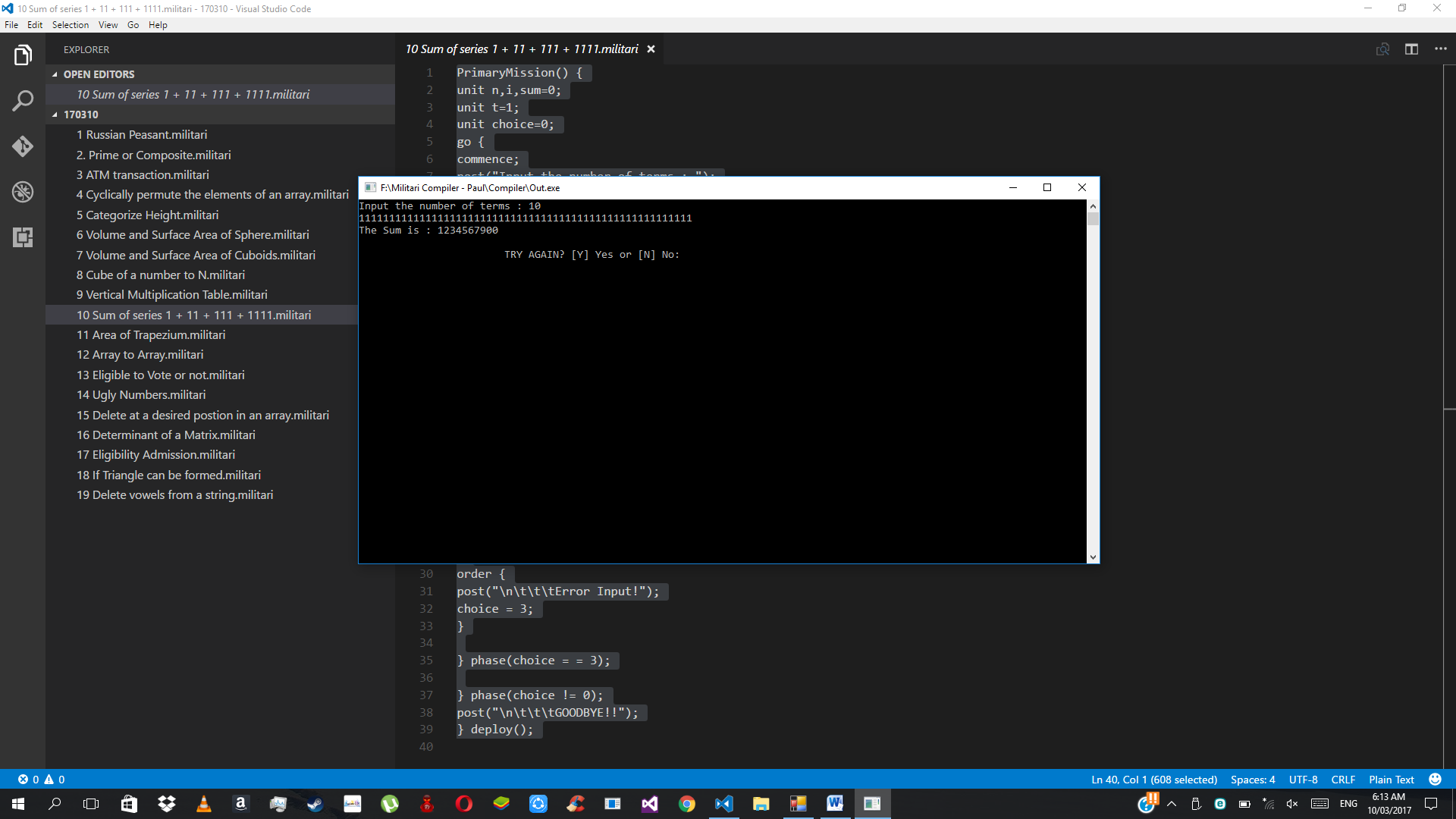
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Sum of series 1 + 11 + 111+ 1111 + …

Sample Input – Output:

.

Source Code:

PrimaryMission() {

unit n,i,sum=0;

unit t=1;

unit choice=0;

go {

commence;

post("Input the number of terms : ");

capture(#n);

inquire(i=1;i<=n;i++) {

post(t);

sum=sum+t;

t=(t\*10) +1;

}

post("\nThe Sum is : " + sum);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

n=0;

i=0;

sum=0;

t=1;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

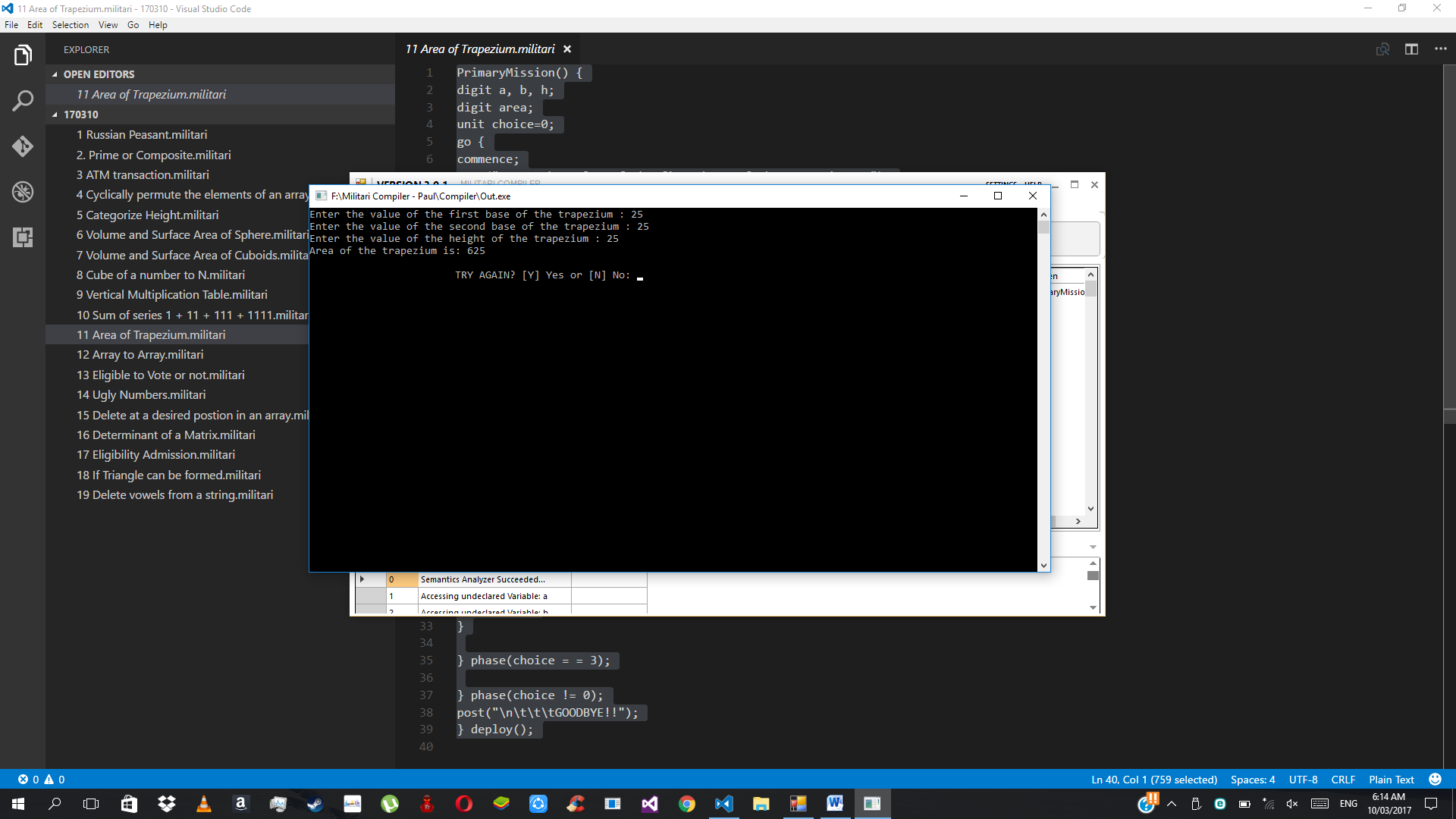
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Area of a Trapezium

Sample Input – Output:



Source Code:

PrimaryMission() {

digit a, b, h;

digit area;

unit choice=0;

go {

commence;

post("Enter the value of the first base of the trapezium : ");

capture(#a);

post("Enter the value of the second base of the trapezium : ");

capture(#b);

post("Enter the value of the height of the trapezium : ");

capture(#h);

area = 0.5 \*(a+b) \*h;

post("Area of the trapezium is: " + area);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

a=0;

b=0;

h=0;

area=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

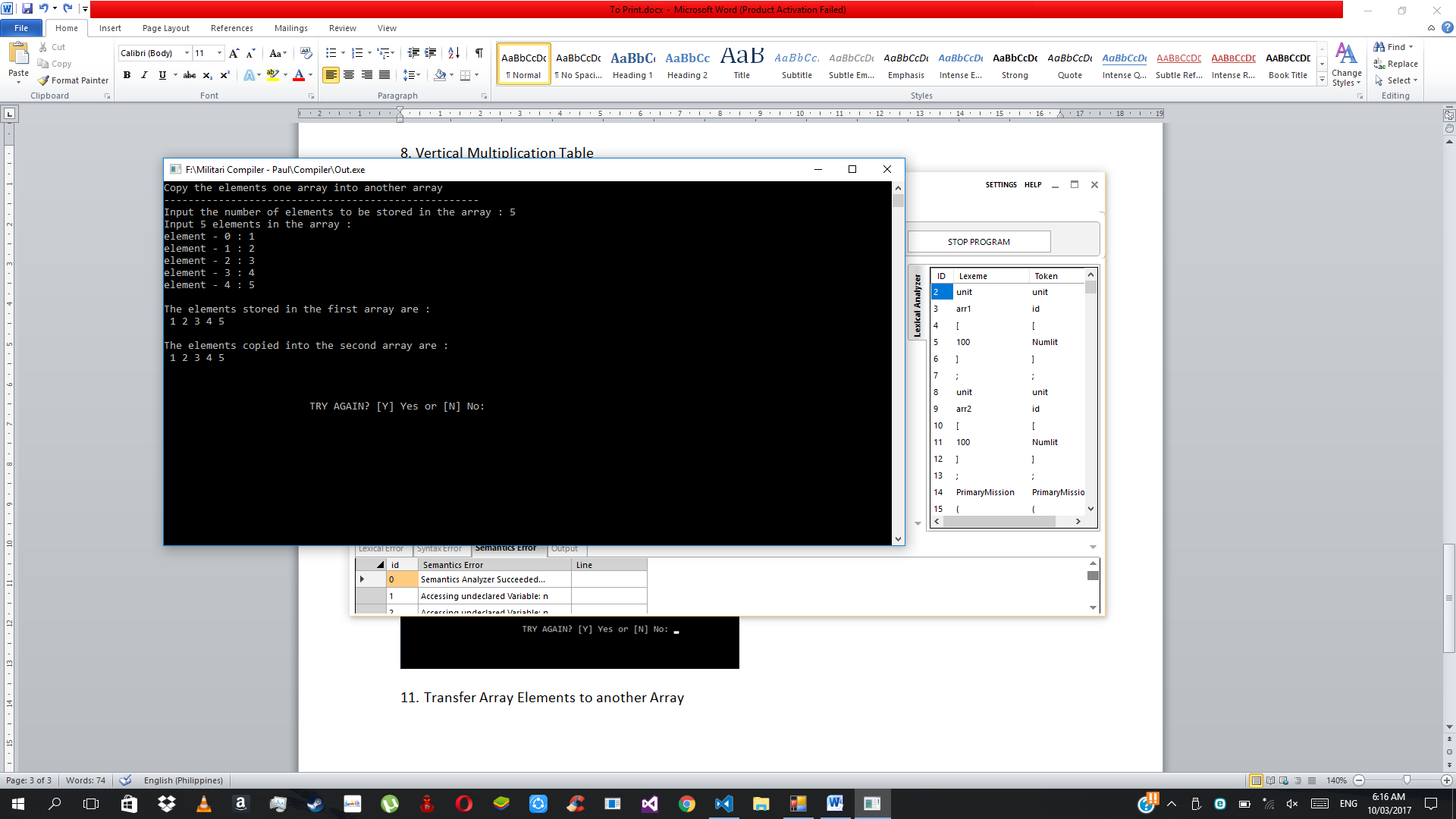
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Transfer Array Elements to another Array

Sample Input – Output:



Source Code:

unit arr1[100];

unit arr2[100];

PrimaryMission() {

unit i, n; unit choice=0;

go { commence; post("Copy the elements one array into another array\n");

post("----------------------------------------------------\n");

post("Input the number of elements to be stored in the array : ");

capture(#n);

post("Input " + n + " elements in the array : \n");

inquire(i=0;i<n;i++) {

post("element - " + i + " : ");

capture(#arr1[i]); }

inquire(i=0; i<n; i++) {

arr2[i] = arr1[i];

} post("\nThe elements stored in the first array are :\n");

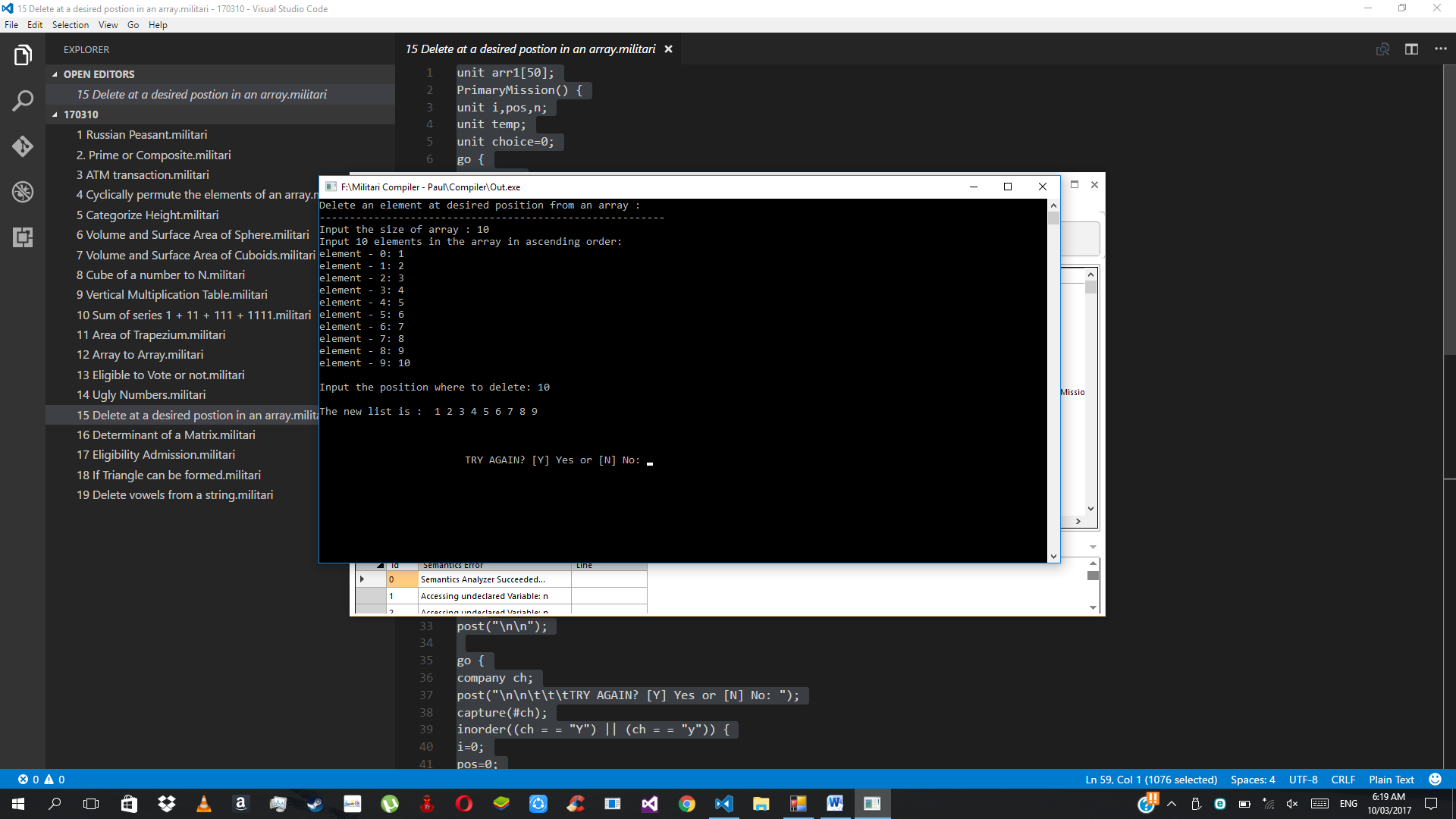
inquire(i=0; i<n; i++) {

post(" " + arr1[i]);

} post("\n\nThe elements copied into

Delete Element from an Array

Sample Input – Output:



Source Code:

unit arr1[50];

PrimaryMission() {

unit i,pos,n; unit temp;

unit choice=0; go {

commence; post("Delete an element at desired position from an array :\n"); post("---------------------------------------------------------\n"); post("Input the size of array : "); capture(#n);

post("Input " + n + " elements in the array in ascending order:\n");

inquire(i=0;i<n;i++) {

post("element - " + i + ": ");

capture(#arr1[i]);

} post("\nInput the position where to delete: ");

capture(#pos); i=0;

temp=pos-1; phase(i !=temp) {

i++;} phase(i<n) {

arr1[i] = arr1[i+1];

i++;} n--;

post("\nThe new list is : ");

inquire(i=0;i<n;i++) {

post(" " + arr1[i]);}

post("\n\n");

go { company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No:");

capture(#ch); inorder((ch = = "Y") || (ch = = "y")) { i=0; pos=0; n=0; temp=0; choice = 1; }

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

} order {

post("\n\t\t\tError Input!");

choice = 3;

} } phase(choice = = 3);

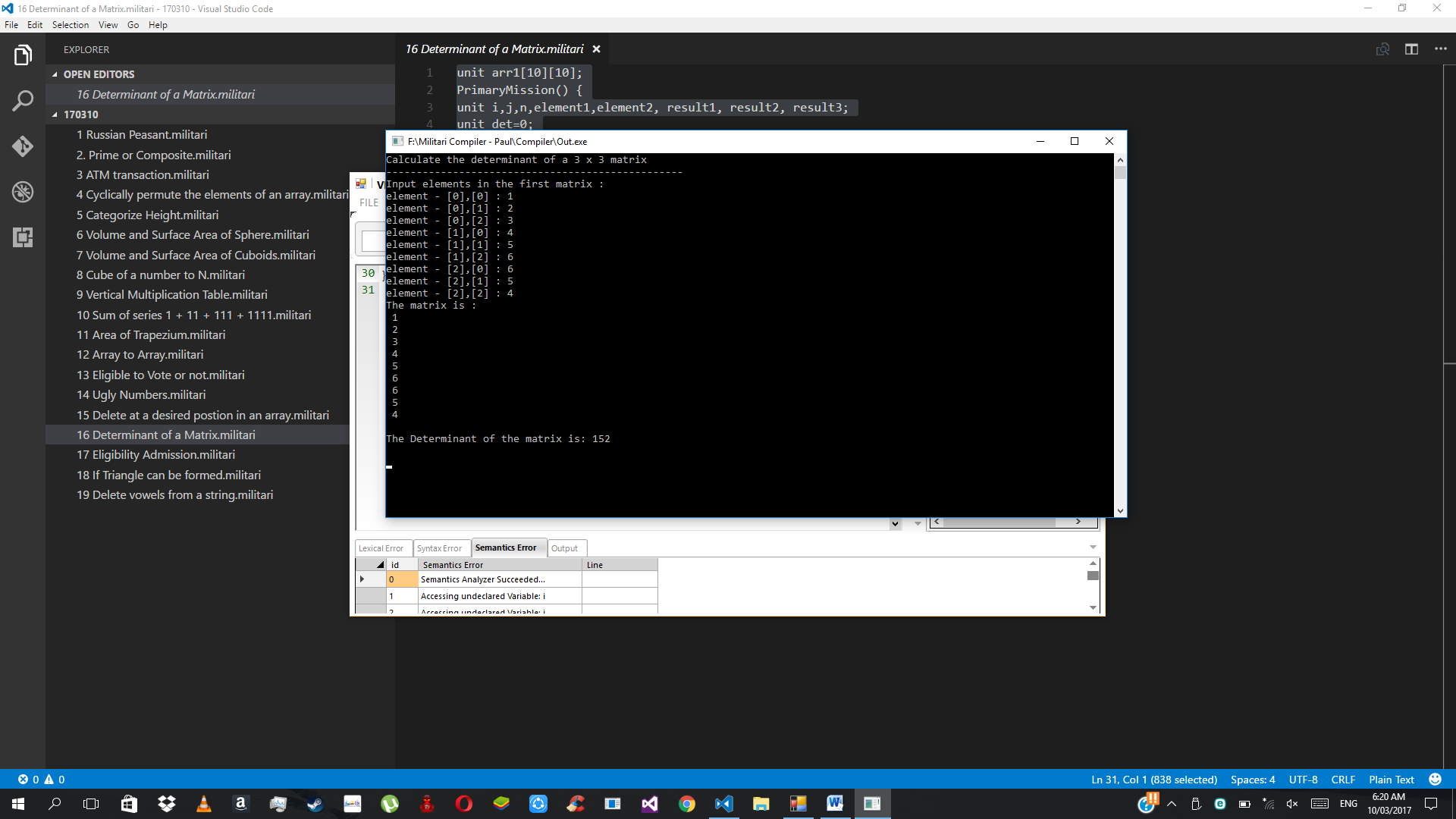
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Determinant of a Matrix

Sample Input – Output:



Source Code:

unit arr1[10][10];

PrimaryMission() {

unit i,j,n,element1,element2, result1, result2, result3;

unit det=0;

post("Calculate the determinant of a 3 x 3 matrix\n");

post("-------------------------------------------------\n");

post("Input elements in the first matrix :\n");

inquire(i=0;i<3;i++) {

inquire(j=0;j<3;j++) {

post("element - [" + i + "],[" + j + "] : ");

capture(#arr1[i][j]);

}

} post("The matrix is :\n");

inquire(i=0;i<3;i++) {

inquire(j=0;j<3 ;j++) {

post(" " + arr1[i][j]);

post("\n"); } }

inquire(i=0;i<3;i++) {

element1 = (i+1) %3;

element2 = (i+2) %3;

result1 = arr1[1][element1] \*arr1[2][element2];

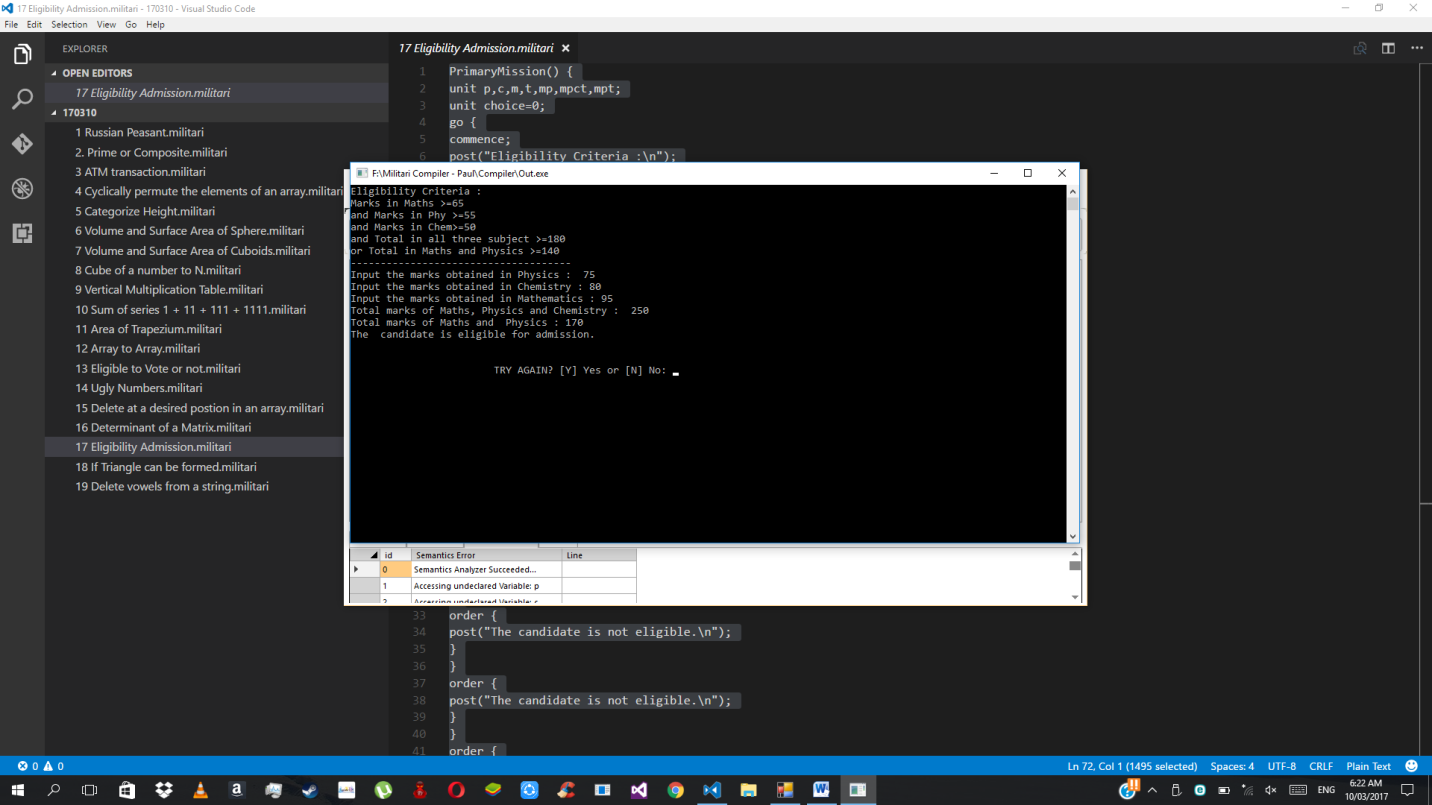
result2 = arr1[1][element2] \*arr1[2][element1];

result3 = result1 - result2;

det = det + (arr1[0][i] \* result2); }post("\nThe Determinant of the matrix is: " + det + "\n\n");

} deploy();

Eligibility for Admission

Sample Input – Output:

Source Code:

PrimaryMission() {

unit p,c,m,t,mp,mpct,mpt;

unit choice=0;

go { commence;

post("Eligibility Criteria :\n");

post("Marks in Maths >=65\n");

post("and Marks in Phy >=55\n");

post("and Marks in Chem>=50\n");

post("and Total in all three subject >=180\n"); post("or Total in Maths and Physics >=140\n"); post("-------------------------------------\n");

post("Input the marks obtained in Physics : "); capture(#p);

post("Input the marks obtained in Chemistry : "); capture(#c);

post("Input the marks obtained in Mathematics : "); capture(#m);

mpct = m+p+c; post("Total marks of Maths, Physics and Chemistry : " + mpct + "\n"); mpt = m+p;

post("Total marks of Maths and Physics : " + mpt + "\n");

inorder(m>=65) { inorder(p>=55) {

inorder(c>=50) { inorder((mpct>=180) || (mpt>=140)) { post("The andidate is eligible for admission.\n"); } order { post("The candidate is not eligible.\n"); } } order { post("The candidate is not eligible.\n"); } } order { post("The candidate is not eligible.\n");}}order { post("The candidate is not eligible.\n"); } go { company ch; post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: "); capture(#ch); inorder((ch = = "Y") || (ch = = "y")) { p=0;c=0;m=0;t=0;mp=0;mpct=0;

mpt=0;choice = 1; }otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;}order { post("\n\t\t\tError Input!");

choice = 3; } phase(choice = = 3);

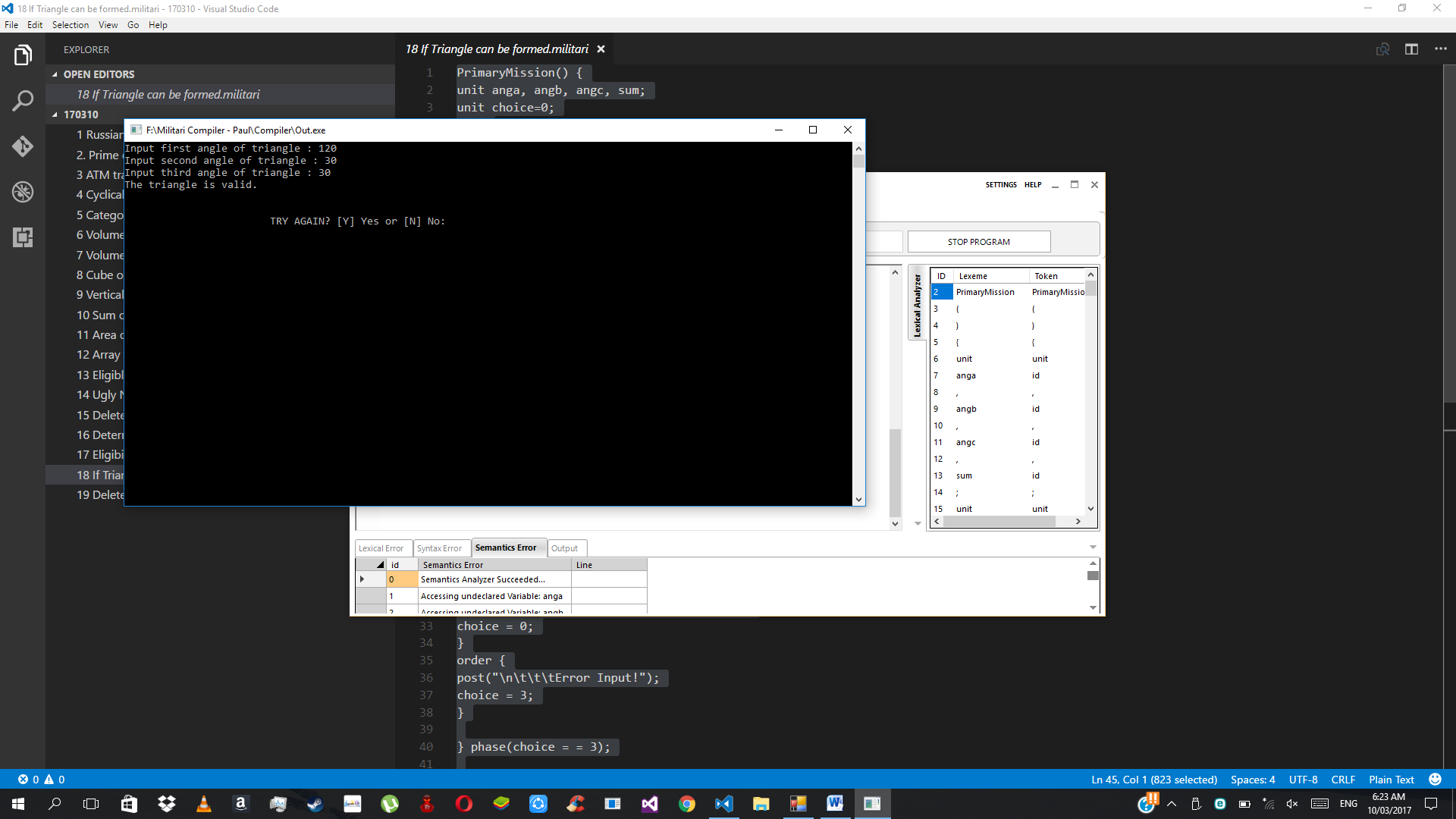
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

If Triangle can be Formed

Sample Input – Output:



Source Code:

PrimaryMission() {

unit anga, angb, angc, sum;

unit choice=0;

go {

commence;

post("Input first angle of triangle : ");

capture(#anga);

post("Input second angle of triangle : ");

capture(#angb);

post("Input third angle of triangle : ");

capture(#angc);

sum = anga + angb + angc;

inorder(sum = = 180) {

post("The triangle is valid.\n");

}

order

{

post("The triangle is not valid.\n");

}

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

anga=0;

angb=0;

angc=0;

sum=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

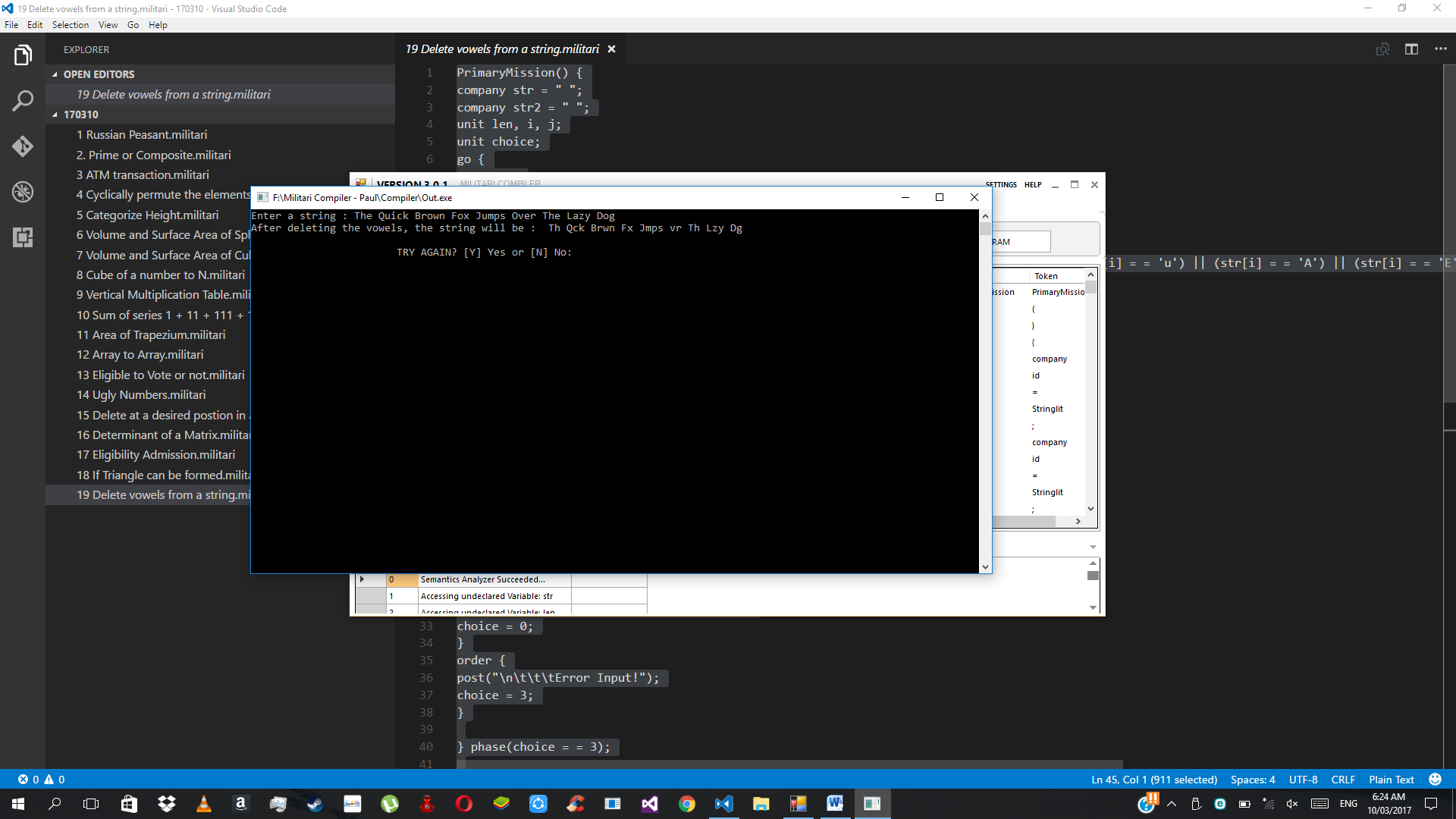
} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();

Delete Vowels from a String

Sample Input – Output:



Source Code:

PrimaryMission() {

company str = " ";

company str2 = " ";

unit len, i, j;

unit choice;

go {

commence;

post("Enter a string : ");

capture(#str);

len=str.Extent;

inquire(i=0; i<len; i++) {

inorder((str[i] = = 'a') || (str[i] = = 'e') || (str[i] = = 'i') || (str[i] = = 'o') || (str[i] = = 'u') || (str[i] = = 'A') || (str[i] = = 'E') || (str[i] = = 'I') || (str[i] = = 'O') || (str[i] = = 'U')) {

}

order {

str2 = str2 + str[i];

}

}

post("After deleting the vowels, the string will be : " + str2);

go {

company ch;

post("\n\n\t\t\tTRY AGAIN? [Y] Yes or [N] No: ");

capture(#ch);

inorder((ch = = "Y") || (ch = = "y")) {

str = " ";

str2 = " ";

len=0;

i=0;

j=0;

choice = 1;

}

otherorder((ch = = "N") || (ch = = "n")) {

choice = 0;

}

order {

post("\n\t\t\tError Input!");

choice = 3;

}

} phase(choice = = 3);

} phase(choice != 0);

post("\n\t\t\tGOODBYE!!");

} deploy();