FAST – National University of Computer & Emerging Sciences

Programming Fundamental Lab

Lab Instructor: Engr. Zakria Bacha

Zakria.bacha@nu.edu.pk

Recursion

When a function call itself is called Recursion. It divides the problem in to sub problems.

Important instruction while solving problem through Recursion:

1. There should be base condition
2. Divide problems in to sub problems

Passing Array in to Function:

Just like we pass variable to function, we can pass array in to function.

```
void fun(char arr[],int size)
{
    cout<<"Printing Array in function";
    for(int i=0;i<size;i++)
    {
        cout<<arr[i]<<" ";
    }

}

int main() {


    char arr[] = "FAST";
    int size = 4;
    fun(arr,4);

return 0;
```

Recursion Example:

```
void fun(int n)
{

    if(n<0)
    return;

    cout<<"Print me";
    fun(n-1);
}


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


    fun(10);

return 0;
```

Pointer:

This type of variable only stores the address of the variable. For example, integer type pointer will store the address of pointer.

Int a = 10;

Int *ptr;

Ptr = &a;

Call by Reference:

```
void fun(int *a,int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;

}
int main() {


    int a=10,b = 20;

    fun(&a,&b);

    cout<< "a Value:"<<a<<endl;
    cout<<"b value:"<<b<<endl;
```

Lab Tasks:

1. Pass character array to a function **"check_size"** and check whether the characters in the array is greater than 4 or not. If greater than 4, then call to another function **"check_reverse",** if the reverse of the characters in the array is same with original array.
2. Write a function "check_start_end" which take two argument the start and end. The function should check if start equal to 2 and end equal to 10 or not. If start =2, and end=10, then find the table of start number till end using recursion.