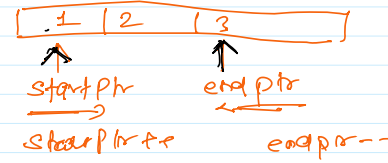
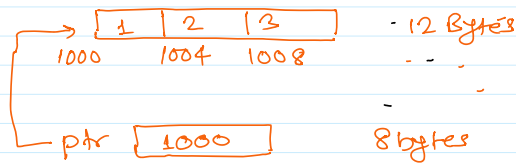


Reverse int array using Pointers

```

int arr [3] = { 1, 2, 3 };
int *startPtr = arr;
int *endPtr = arr + 2;
int temp;

```



```

for (int i=0; i<3/2; i++)
{

```

```

    temp = *startPtr; // 1

```

```

    *startPtr = *endPtr;

```

```

    *endPtr = temp;

```

```

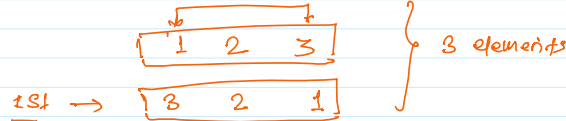
    startPtr++;

```

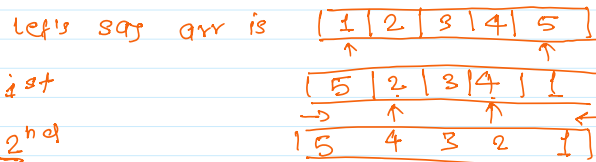
```

    endPtr--;
}

```



$$\text{Iteration} = \text{No. of elements} / 2 = 3 / 2 = 1$$



$$\text{Iteration} = \text{No. of ele} / 2 = 5 / 2 = 2$$

```

arr [1 2 3]
    1000 1004 1008

```

```

startPtr [1000]

```

```

endPtr [1008]

```

Example :-

```

arr [1 2 3 4 5]
    1000 1004 1008 1012 1016

```

```

int temp;

```

```

startPtr = arr;

```

```

endPtr = arr + 4;

```

```

for (int i=0; i<5/2; i++)
{

```

```

    temp = *startPtr;
    *startPtr = *endPtr;
    *endPtr = temp;

```

```

    startPtr++;

```

```

    endPtr--;
}

```

|| 0 1

5/2 = 2 times loop

```

startPtr [1008]

```

```

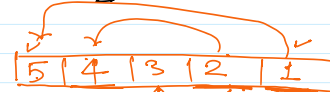
endPtr [1008]

```

```

temp [2]

```



```

}

```

```

i=0    i<5/2    0<2 True
i=1    1<5/2    1<2 True
i=2    2<5/2    2<2 False

```

```

// print reverse array

```

```

for (int j=0; j<5; j++)
{

```

```

    printf ("%d ", arr[j]);

```

```

}

```

```

// Lets print value of startPtr & endPtr ??

```

```

printf ("%d", *startPtr); // 3

```

```
printf("%d", *endptr);    173  
printf("%d", temp);      172
```

= x =