

## Tutorial 1 – solutions

Q1) Which of the following is true about new when compared with malloc.

- 1) new is an operator, malloc is a function
  - 2) new calls constructor, malloc doesn't
  - 3) new returns appropriate pointer, malloc returns void \* and pointer needs to typecast to appropriate type.
- a) 1 & 2
  - b) 1 & 3
  - c) 2 & 3
  - d) All 1, 2 & 3

Q2) Array p: 5 7 11 17 25  
Array q: 25 17 11 7 5

Q3) Write a C++ program to perform Insertion and deletion operation in array by creating array using dynamic memory allocation (new and delete operator).

```
#include <iostream>
using namespace std;
int main()
{
    int len, sum = 0;
    cout<< "Enter the no. of students in the class" <<endl;
    cin>>len;
    int *marks = new int[len]; //Dynamic memory allocation
    cout<< "Enter the marks of each student" <<endl;
    for( int i = 0; i<len; i++ )
    {
        cin>> marks[i];
    }
    for( int i = 0; i<len; i++ )
    {
        sum += marks[i];
    }
    cout<< "sum is " << sum <<endl;
    delete[]marks;
    return 0;
}
```

Q4) Write a C++ program to create a new array that is twice the size of the argument array. The function should copy the contents of the argument array to the first half of the new array, and the contents of the argument array each multiplied by 2 to the second half of the new array. The function should return a pointer to the new array

```

int *doubleArray(int *array, int size)
{
if (size <= 0)
return NULL;
int *newArray = new int[size * 2];
int i = 0;
// first half
for (i = 0; i < size; i++)
{
*(newArray + i) = *(array + i);
}
// second half
for (i = size; i < size * 2; i++)
{
*(newArray + i) = *(array + i - size) * 2;
}
return newArray;
}

```

Q5)

Sample Code

```

#include <iostream>
using namespace std;

//Declare Node
struct Node{
intnum;
    Node *next;
};

//Declare starting (Head) node
struct Node *head=NULL;

//Insert node at start
voidinsertNode(int n){
struct Node *newNode=new Node;
newNode->num=n;
newNode->next=head;
    head=newNode;
}

//Traverse/ display all nodes (print items)
void display(){

```

```

if(head==NULL){
cout<<"List is empty!"<<endl;
    return;
}
    struct Node *temp=head;
while(temp!=NULL){
cout<<temp->num<<" ";
    temp=temp->next;
}
cout<<endl;
}

//delete node from start
voiddeleteItem(){
if(head==NULL){
cout<<"List is empty!"<<endl;
    return;
}
cout<<head->num<<" is removed."<<endl;
    head=head->next;
}
int main(){

display();
insertNode(10);
insertNode(20);
insertNode(30);
insertNode(40);
insertNode(50);
display();
deleteItem(); deleteItem(); deleteItem(); deleteItem(); deleteItem();
deleteItem();
display();
    return 0;
}

```

### Output

