***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B11***

***Data Structure Lab – II [15B17CI271]***

***Lab A Week 2***

A.

#include <iostream>

//#pragma pack(1)

using namespace std;

class abc

{

int x;

double y;

};

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2 : "<<sizeof(o2);

cout<<"\nSize of abc is :"<<sizeof(abc);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2 : 16

Size of abc is : 16

without padding -

Size of o1 : 8

Size of o2 : 12

Size of abc is : 12

B.

#include <iostream>

using namespace std;

class abc

{

int x;

double y;

int z;

};

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2 : "<<sizeof(o2);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2 : 24

without padding -

Size of o1 : 8

Size of o2 : 16

C.

#include <iostream>

using namespace std;

class abc

{

int x;

double y;

int z;

int a;

};

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2 : "<<sizeof(o2);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2: 24

without padding -

Size of o1 : 8

Size of o2 : 20

D.

#include <iostream>

using namespace std;

class abc

{

float x;

char y;

int z;

double a;

};

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2 : "<<sizeof(o2);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2 : 24

without padding -

Size of o1 : 8

Size of o2 : 17

E.

#include <iostream>

using namespace std;

class abc

{

char x[5];

double y;

}

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2: "<<sizeof(o2);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2 : 16

without padding -

Size of o1 : 8

Size of o2 : 13

F.

#include <iostream>

using namespace std;

class abc

{

char x[5];

float y[3];

};

int main()

{

abc \*o1 = new abc, o2;

cout<<"\nSize of o1 : "<<sizeof(o1);

cout<<"\nSize of o2 :"<<sizeof(o2);

return 0;

}

output : with padding -

Size of o1 : 8

Size of o2 : 20

without padding -

Size of o1 : 8

Size of o2 : 17

A.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

float \*a;

a = (float \*)malloc(sizeof(int));

a[0] = 4.5;

cout<<a[0];

return 0;

}

output : 4.5

B.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a;

a = (int \*)malloc(sizeof(float));

a[0] = 5;

cout<<a[0];

return 0;

}

output : 5

C.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a, \*b;

a = (int \*)malloc(sizeof(int));

b = (int \*)malloc(5\*sizeof(int));

cout<<sizeof(a)<< sizeof(b);

return 0;

}

output : 88

D.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a;

a = (int \*)malloc(sizeof(int));

a[0] = 5;

cout<<a[0];

return 0;

}

output : 5

E.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a[5];

a[0] = (int \*)malloc(sizeof(int));

a[0][0] = 5;

cout<<a[0][0];

return 0;

}

output : 5

F.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

struct node

{

int a[10];

};

struct node \*n;

n = (struct node \*)malloc(sizeof(struct

node));

cout<<sizeof(n);

return 0;

}

output : 8

G.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a[5];

a[0] = (int

\*)malloc(2\*sizeof(int));

a[0][1]

= 5;

cout<<a[0][1];

return 0;

}

output : 5

H.

#include <iostream>

#include <malloc.h>

using namespace std;

int main()

{

int \*a = (int \*)malloc(5\*sizeof(int));

a[0] = 1;

a[1] = 2;

a[2] = 3;

a[3] = 4;

a[4] = 5;

delete(a);

cout<<a[0]<<a[1]<<a[2]<<a[3]<<a[4];

return 0;

}

output : garbage value

A dynamically created array stores following integer elements (odd and even integers)

2 8

3 6 7 9 5 4

It is desired to print/display the elements of this array in such manner that it first prints all the even

elements then it prints all the odd elements. In above example, the displayed elements are as follows:

2 8 6 4 3 7 9 5

Write a program with and without STL to create the dynamic array of user inputted length (n), assign

values at different indices of the array, and as presented in above example, display the elements of this

array.

(Note: don’t enter the elements manually, rather use following statement in loop to randomly assign

elements (in range between 0 and 99) in the array: A[i] = rand()%100, where A is an array).

without STL

#include <iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int \*arr = new int[n];

for(int i=0; i<n; i++)

{

arr[i] = rand()%100;

}

for(int i=0; i<n; i++)

{

if(arr[i] % 2 == 0)

{

cout<<arr[i]<<" ";

}

}

for(int i=0; i<n; i++)

{

if(arr[i] % 2 != 0)

{

cout<<arr[i]<<" ";

}

}

return 0;

}

with STL

#include <bits/stdc++.h>

using namespace std;

int main()

{

vector<int> v;

int n;

cin >> n;

int \*arr = new int[n];

for (int i = 0; i < n; i++)

{

arr[i] = rand() % 100;

}

for (int i = 0; i < n; i++)

{

if (arr[i] % 2 == 0)

{

v.push\_back(arr[i]);

}

}

for (int i = 0; i < n; i++)

{

if (arr[i] % 2 != 0)

{

v.push\_back(arr[i]);

}

}

for(auto x : v)

{

cout<<x<<" ";

}

return 0;

}