LAB 10 TASKS (POINTERS)

Q.1) What will be the output of the following program?

```
#include
<iostream> using
namespace std;

int main()
{
    int a = 32, *ptr =
    &a; char ch = 'A',
    &cho = ch;

    cho += a;
    *ptr += ch;
    cout << a << ", " << ch <<
    endl; return 0;
}
```

Output:

```
129, a

--------------------------------
Process exited after 0.08485 seconds with return value 0
Press any key to continue . . .
```

Step-by-Step Explanation:

1. **Initializations**:
   - `int a = 32;`
     Variable `a` is assigned the value `32`.
   - `int *ptr = &a;`
     Pointer `ptr` is initialized to store the address of `a`.
   - `char ch = 'A';`
     Character `ch` is assigned the value `'A'`, whose ASCII value is `65`.
   - `char &cho = ch;`
     `cho` is declared as a reference to `ch`, meaning any changes to `cho` will directly affect `ch`.

2. **`cho += a;`**
   - `cho` refers to `ch`, so this operation modifies `ch`.
   - The current value of `a` is `32`.
   - `cho = 'A' + 32 = 65 + 32 = 97`, which corresponds to the character `'a'`.

3. **`*ptr += ch;`**
   - `ptr` points to `a`, so `*ptr` represents the value of `a`.
   - The updated value of `ch` (or `cho`) is `'a'` (ASCII value = `97`).
   - `*ptr = a + ch = 32 + 97 = 129`.
   - This means the value of `a` is now updated to `129`.

4. `cout << a << ", " << ch << endl;`
    - The value of `a` is `129` after the above operations.
    - The value of `ch` is `'a'`.

Q.2) Write a C++ program to reverse an array using pointers

Source Code:

```cpp
#include <iostream>
using namespace std;

void reverseArray(int *arr, int size) {
    int *start = arr;
    int *end = arr + size - 1;

    while (start < end) {
        int temp = *start;
        *start = *end;
        *end = temp;

        start++;
        end--;
    }
}

int main() {
    int n;
    cout << "Enter the size of the array: ";
    cin >> n;
    int arr[n];
    cout << "Enter " << n << " elements of the array:\n";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    reverseArray(arr, n);
    cout << "Reversed array:\n";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Output:

```
Enter the size of the array: 5
Enter 5 elements of the array:
5
6
7
8
9
Reversed array:
9 8 7 6 5

--------------------------------
Process exited after 4214 seconds with return value 0
Press any key to continue . . .
```

DOUBLE POINTERS

Tasks

1. Correct errors if any and Print Output

a)

```
#include<iostream>
using namespace std;
main()
{
    float num = 10;
    float *pt1 = &num;
    float **pt2 = &pt1;
    cout<<" address of num =  "<<pt1<<endl;
    cout<<" address stored by pt1 = "<<&num<<endl;
    cout<<" value pointed by pt1 = "<<*pt1<<endl;
    cout<<" address of pt1 = "<<pt2<<endl;
    cout<<" address stored by pt2 = "<<&pt1<<endl;
    cout<<" value of num  = "<<**pt2;
}
```

Errors:

1. The main() function should explicitly return an integer (int main()).
2. The first cout statement incorrectly prints the value of pt1 (which is the address of num) instead of the address of num itself (&num).
3. The second cout statement incorrectly prints &num (which is the address of num) instead of the value stored in pt1 (which is also the address of num).

Corrected Code

```
#include<iostream>
using namespace std;
int main()
{
    float num = 10;
    float *pt1 = &num;
    float **pt2 = &pt1;
    cout << " address of num = " << &num << endl;
    cout << " address stored by pt1 = " << pt1 << endl;
    cout << " value pointed by pt1 = " << *pt1 << endl;
    cout << " address of pt1 = " << &pt1 << endl;
    cout << " address stored by pt2 = " << pt2 << endl;
    cout << " value of num = " << **pt2 << endl;
    return 0;
}
```

Output

```
address of num = 0x23fe44
address stored by pt1 = 0x23fe44
value pointed by pt1 = 10
address of pt1 = 0x23fe38
address stored by pt2 = 0x23fe38
value of num = 10

--------------------------------
Process exited after 0.08031 seconds with return value 0
Press any key to continue . . .
```

Explanation of Corrections:

1. Changed main() to int main() and added return 0; to comply with C++ standards.
2. The first cout now correctly prints the address of num using &num.
3. The second cout now correctly prints the address stored in pt1 (which is the address of num).
4. The fourth cout now correctly prints the address of pt1 using &pt1.
5. The fifth cout now correctly prints the address stored in pt2 (which is the address of pt1).

b)

```
#include<iostream>
using namespace std;
main()
{
    char *p1;
    int a = 8;
    p1 = &a;
    cout<<p1;
}
```

Errors:

1. The main() function should explicitly return an integer (int main()).
2. There is a type mismatch: p1 is a char*, but &a is an int*. Assigning an int* to a char* is not allowed without a cast.
3. Printing p1 as a char* will not produce meaningful output because it points to an integer, not a string.

Source Code

```
#include<iostream>
using namespace std;
int main()
{
    char *p1;
    int a = 8;
    p1 = (char*)&a;  // Casting to char* to avoid type mismatch
    cout << p1 << endl;
    return 0;
}
```

Output

```
--------------------------------
Process exited after 0.05281 seconds with return value 0
Press any key to continue . . .
```

Explanation of Corrections:

1. Changed main() to int main() and added return 0; to comply with C++ standards.
2. Added a cast (char*) to convert the int* to a char*. This resolves the type mismatch.
3. The output will be non-printable characters or garbage because p1 points to an integer, not a string.

c)

```
#include<iostream>
using namespace std;
main()
{
    int a =5;
    int *p = &a;
    cout<<++*p<<endl;
    cout<<*p++;
}
```

Errors:

1. The main() function should explicitly return an integer (int main()).
2. The second cout statement has a syntax error: "p++ is missing a closing quote and a semicolon.
3. The second cout statement should print the value of *p after incrementing the pointer.

Source Code

```
#include<iostream>
using namespace std;
int main()
{
    int a = 5;
    int *p = &a;
    cout << ++*p << endl;
    cout << *p++ << endl;
    return 0;
}
```

Output

```
6
6

--------------------------------
Process exited after 0.05104 seconds with return value 0
Press any key to continue . . .
```

Explanation of Corrections:

1. Changed main() to int main() and added return 0; to comply with C++ standards.
2. Fixed the syntax error in the second cout statement by removing the extra quote and adding a semicolon.
3. The second cout now correctly prints the value pointed by p and then increments the pointer.

d)

```
#include<iostream>
using namespace std;
main()
{
    int a =5;
    int *p = &a;
    cout<<p<<endl;
    cout<<++*p<<endl;
    cout<<*p++;
    cout<<p;
}
```

Errors:

1. The main() function should explicitly return an integer (int main()).
2. The second cout statement has a syntax error: "++*p << endl; is missing a closing quote.
3. The third cout statement has a syntax error: "p++; is missing a closing quote and a semicolon.
4. The third cout statement should print the value of *p after incrementing the pointer.

Source Code

```
#include<iostream>
using namespace std;
int main()
{
    int a = 5;
    int *p = &a;
    cout << p << endl;
    cout << ++*p << endl;
    cout << *p++ << endl;
    cout << p << endl;
    return 0;
}
```

Output

```
0x23fe44
6
6
0x23fe48

--------------------------------
Process exited after 0.09568 seconds with return value 0
Press any key to continue . . .
```

Explanation of Corrections:

1. Changed main() to int main() and added return 0; to comply with C++ standards.
2. Fixed the syntax error in the second cout statement by removing the extra quote.
3. Fixed the syntax error in the third cout statement by removing the extra quote and adding a semicolon.
4. The third cout now correctly prints the value pointed by p and then increments the pointer.
5. The fourth cout prints the new address stored in p after incrementing.