BÀI TẬP TUẦN 2 – CON TRỎ

1. Phát biểu nào sau đây cho giá trị được lưu trữ tại địa chỉ được trỏ bởi con trỏ a

a) a

b) val(a)

c) \*a //Toán tử \* truy xuất đến giá trị của địa chỉ

d) &a

2. Phát biểu nào sau đây là có từ khoá đúng để giải phóng bộ nhớ trong C

a) delete

b) free //Dùng free() để giải phóng bộ nhớ trong C

c) clear

d) remove

3. Kiểu trả về của hàm malloc() hoặc calloc() là gì?

a) void\* //Malloc() hoặc calloc() trả về void\*

b) con trỏ của kiểu bộ nhớ được cấp phát

c) void\*\*

d) int\*

4. Vấn đề gì xảy ra với chương trình sau

#include<stdio.h>

int main()

{

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

p = NULL;

free(p);

}

a) Lỗi biên dịch

b) Thiếu bộ nhớ // Vì p được cấp phát bộ nhớ không được giải phóng, nhưng con trỏ được gán là NULL => sai trình tự (free(p) -> p=NULL)

c) Lỗi thời gian thực thi

d) Chương trình ngừng khi hàm free() được gọi cho con trỏ NULL.

5. Đầu ra của chương trình sau là gì?

# include <stdio.h>

void fun(int x)

{

x = 30;

}

int main()

{

int y = 20;

fun(y);

printf("%d", y);

return 0;

}

a) 30

b) 20 // Tham số hàm fun ở dạng tham trị nên khi truyền y vào hàm giá trị bên ngoài của y vẫn không bị ảnh hưởng khi thay đổi giá trị bên trong hàm

c) Lỗi biên dịch

d) Lỗi thời gian thực thi

6. Đầu ra của chương trình sau là gì?

# include <stdio.h>

void fun(int \*ptr)

{

\*ptr = 30;

}

int main()

{

int y = 20;

fun(&y);

printf("%d", y);

return 0;

}

a) 20

b) 30 // Vì chương trình gọi hàm fun với tham chiếu (&y), lúc này sẽ in ra địa chỉ vùng nhớ của biến con trỏ \*ptr => kết quả là 30

c) Lỗi biên dịch

d) Lỗi thời gian thực thi

7. Đầu ra của chương trình sau là gì?

#include<stdio.h>

void f(int \*p, int \*q) // p trỏ đến i và q trỏ đến j

{

p = q; // p trỏ đến j

\*p = 2; // giá trị của j được gán bằng 2

}

int i = 0, j = 1;

int main()

{

f(&i, &j);

printf("%d %d \n", i, j);

getchar();

return 0;

}

a) 2 2

b) 2 1

c) 0 1

d) 0 2

8. Đầu ra của chương trình sau là gì?

#include <stdio.h>

int main()

{

int arr[] = {1, 2, 3, 4, 5};

int \*p = arr;

++\*p;

p += 2;

printf("%d", \*p);

return 0;

}

a) 2

b) 3 // ++ \* p => “++ (\* p)”. Nó tăng giá trị của phần tử đầu tiên của mảng (không thay đổi con trỏ p).

Khi p + = 2 được thực hiện, p được thay đổi để trỏ đến phần tử thứ ba của mảng.

c) 4

d) Lỗi biên dịch

9. Đầu ra chương trình sau là gì?

#include<stdio.h>

int main()

{

int a;

char \*x;

x = (char \*) &a;

a = 512;

x[0] = 1;

x[1] = 2;

printf("%d\n",a);

return 0;

}

a) Phụ thuộc vào máy // a=512=0x00000200 => giá trị của các byte từ 1-4 lưu biến a

b) 513

c) 258

d) Lỗi biên dịch

10. Đầu ra chương trình sau là gì?

#include<stdio.h>

int main()

{

int a = 12;

void \*ptr = (int \*)&a;

printf("%d", \*ptr); // Lỗi, phải ép kiểu trước khi lấy ra giá trị mà con trỏ trỏ đến (printf("%d", \*(int \*)ptr);

getchar();

return 0;

}

a) 12

b) Lỗi biên dịch

c) Lỗi thời gian thực thi

d) 0

11. Đầu ra chương trình sau là gì?

#include <stdio.h>

int main()

{

int ary[4] = {1, 2, 3, 4};

int \*p = ary + 3; // con trỏ p giữ vị trí phần tử thứ 4

printf("%d , %d\n", p[-2], ary[\*p]); // => p[-2] có giá trị của phần tử thứ 2 bằng 2

}

a) 2 , 3

b) Lỗi thời gian biên dịch

c) 2 , 4

d) 2 , giá trị rác

12. Đầu ra chương trình sau là gì?

#include <stdio.h>

void main()

{

char \*s= "Computer";

char \*p = s;

printf("%c\t%c", \*(p + 3), s[0]); // \*(p+3) trả về ký tự thứ 4 (p), s[0] trả về C

}

(a) C o

(b) p C

(c) m p

(d) o u