14. The following three files are compiled using the command
`$> gcc main.c prog1.c prog2.c`
What is the output of the executable code?

```
File <main.c>
---------------------
#include <stdio.h>
int x=0;
void main() {
  foo();
  fun();
  printf("Main: %d", x);
}
File <foo.c>
---------------------
#include <stdio.h>
static int x=10;
void foo() {
  x++;
  printf("foo: %d, ", x);
}
File <fun.c>
---------------------
#include <stdio.h>
int x=5;
void fun() {
  x++;
  printf("fun: %d, ", x);
}
```

foo 11

(A) `foo: 11, fun: 1, Main: 1`
(B) `foo: 11, fun: 6, Main: 6`
[✓ (C) Compilation error]
(D) Runtime error

15. In the context of process state, find the odd one out:
(A) Ready
(B) Waiting
[✓ (C) Stored]
(D) Terminated

16. Find the odd one out.
[✓ (A) Motherboard]
(B) CPU
(C) Hard disk
(D) RAM

17. What is the output of the following program?

```
#include <stdio.h>
void outfoo(int k) {
  static t = 0;
  t++;
  printf(" %d ",t);
  t=k;
}
void main() {
  int num=4;
  void (*foo)(int x);

  foo = outfoo;
  foo(num);
  (*foo)(8);
}
```

(A) 4 8
(B) 5 9
[✓ (C) 1 5]
(D) 1 2

1

18. Find the best set:
(A) stack, heap, code, data
(B) stack, heap, code, array
(C) char, pointer, data, stack
(D) array, stack, data, code

19. English like statements are written in
(A) Machine language
(B) Assembly language
[✓ (C) C-program]
(D) All of the above

20. Two object modules have the following goto statements:
Module-A– `A: goto 100;`
Module-B– `B: goto 200;`
At link time the base addresses of the modules are: Module-A=100, Module-B=350. After linking the goto statements will get rewritten as:

(A) `A: goto 100; B: goto 200;`
(B) `A: goto 200; B: goto 350;`
(C) `A: goto 200; B: goto 550;`
(D) `A: goto 200; B: goto 500;`