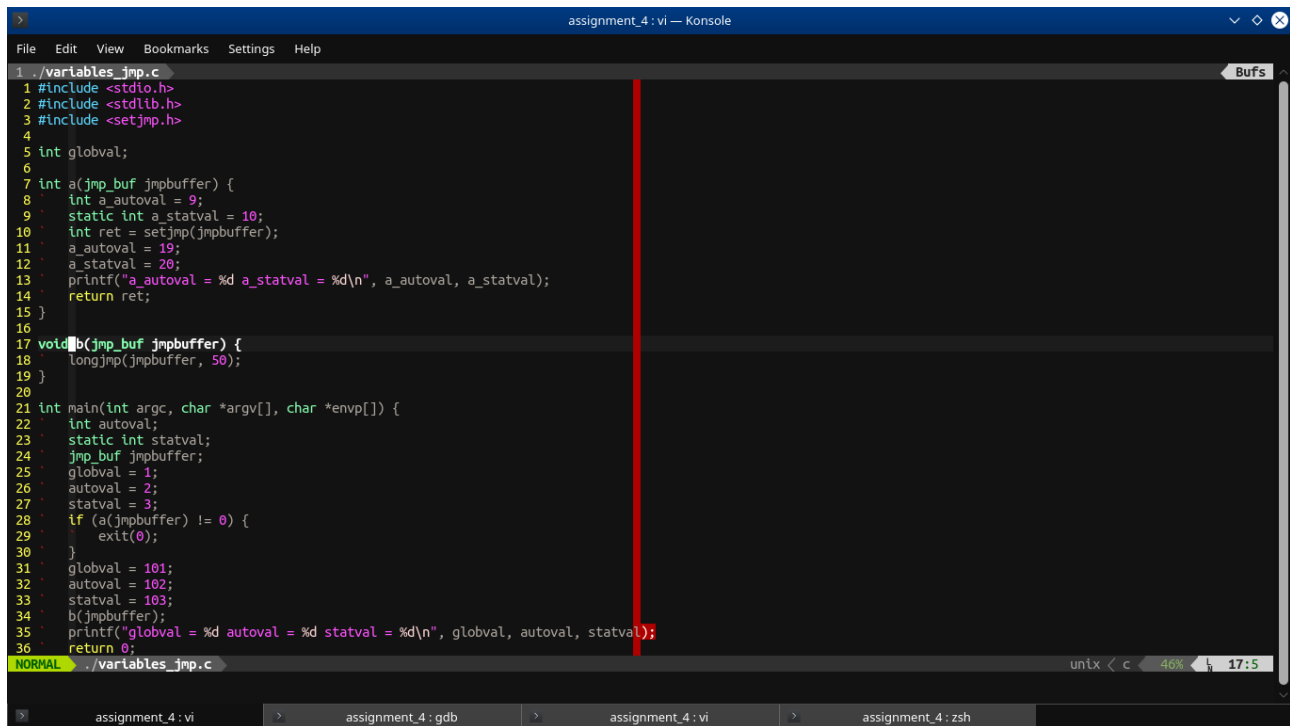


## Assignment 4

Srishti Shelke 111603056  
Niramay Vaidya 111605075

1. Write a program to include different types of variables to demonstrate the behaviour of setjmp/longjmp. Describe your observation and understanding.

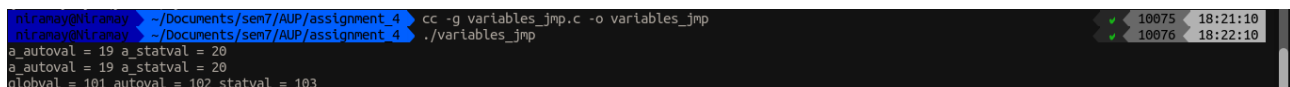
Code-



```
1 ./variables_jmp.c
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <setjmp.h>
5
6 int globval;
7
8 int a(jmp_buf jmpbuffer) {
9     int a_autoval = 9;
10    static int a_statval = 10;
11    int ret = setjmp(jmpbuffer);
12    a_autoval = 19;
13    a_statval = 20;
14    printf("a_autoval = %d a_statval = %d\n", a_autoval, a_statval);
15    return ret;
16 }
17
18 void b(jmp_buf jmpbuffer) {
19     longjmp(jmpbuffer, 50);
20 }
21
22 int main(int argc, char *argv[], char *envp[]) {
23     int autoval;
24     static int statval;
25     jmp_buf jmpbuffer;
26     globval = 1;
27     autoval = 2;
28     statval = 3;
29     if (a(jmpbuffer) != 0) {
30         exit(0);
31     }
32     globval = 101;
33     autoval = 102;
34     statval = 103;
35     b(jmpbuffer);
36     printf("globval = %d autoval = %d statval = %d\n", globval, autoval, statval);
37     return 0;
38 }
```

**Compiled without optimization-** in this case, global, automatic and static variables are all stored in memory and they have values stored in them as they were at the time of longjmp.  
(**Doubt-** why does the problem that occurs while compiling using optimization as described below not get reflected in this case?)

Output-



```
niranay@niranay:~/Documents/sem7/AUP/assignment_4$ cc -g variables_jmp.c -o variables_jmp
niranay@niranay:~/Documents/sem7/AUP/assignment_4$ ./variables_jmp
a_autoval = 19 a_statval = 20
globval = 101 autoval = 102 statval = 103
```

**Compiled with optimization-** in this case, global and static variables stay in memory but automatic variables go to registers and they get restored to their previous values when setjmp was called, in contrast to those in memory which will have values as of the time of longjmp. But here, a segmentation fault occurs. The reason being one can only longjmp() back up the stack. In this code, longjmp() is done in b() and setjmp() is in a() which is down the stack from main(). Hence, the stack frame referenced by jmpbuffer no longer exists after the longjmp() call is made due to which the debugger output after the return function of a() post longjmp() states cannot find bounds of current function. According to longjmp() documentation, the routine may not be called after the routine which called the setjmp() routine returns. This includes returning through a longjmp() out of the function. Also, the local automatic variable's value in a() does not get restored to the previous value in spite of optimized compilation since the stack frame of a() no longer exists.

## Output-

```
niranay@niranay: ~/Documents/sem7/AUP/assignment_4 $ cc -O -g variables_jmp.c -o variables_jmp
niranay@niranay: ~/Documents/sem7/AUP/assignment_4 $ ./variables_jmp
a_autoval = 19 a_statval = 20
a_autoval = 19 a_statval = 20
[1] 29131 segmentation fault (core dumped) ./variables_jmp
```

2. Print all existing environment variables with their values. Later input a new variable and its value and add to the environment list. Once again print the list.

## Code-

```
assignment_4: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./print_env_var.c
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <unistd.h>
6 #include <errno.h>
7
8 int main(int argc, char *argv[], char *envp[]) {
9     if (argc != 4) {
10         printf("Usage ./print_env_var <env_var_name> <env_var_value> <rewrite (Y/N)>\n");
11         exit(0);
12     }
13     /* ./print_env_var name value
14      * ./print_env_var temp name
15      * the code below does not add the above key value pair to the system env
16      * vars since executing env | grep "name"/"temp"/"value" outputs nothing
17      */
18     int i;
19     for (i = 0; envp[i] != NULL; i++) {
20         printf("%s\n", envp[i]);
21     }
22     /* TODO: Find out why using only argv[1] and argv[2] below does not work
23      * with strdup, string is name=value
24      * without strdup, string is name=
25      */
26     envp[i] = strcat(strcat(strdup(argv[1]), "="), strdup(argv[2]));
27     envp[i + 1] = NULL;
28     for (i = 0; envp[i] != NULL; i++) {
29         printf("%s\n", envp[i]);
30     }
31     /* man 7 environ
32      * GNU_SOURCE not defined
33      * cpp -dM /dev/null | grep "_GNU_SOURCE" outputs nothing
34      */
35     extern char **environ;
36     int i, rewrite;
37     NORMAL ./print_env_var.c
38     unix < utf-8 < c 56% 36:19
```

```
assignment_4: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./print_env_var.c
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <unistd.h>
6 #include <errno.h>
7
8 int main(int argc, char *argv[], char *envp[]) {
9     if (argc != 4) {
10         printf("Usage ./print_env_var <env_var_name> <env_var_value> <rewrite (Y/N)>\n");
11         exit(0);
12     }
13     /* ./print_env_var name value
14      * ./print_env_var temp name
15      * the code below does not add the above key value pair to the system env
16      * vars since executing env | grep "name"/"temp"/"value" outputs nothing
17      */
18     int i;
19     for (i = 0; envp[i] != NULL; i++) {
20         printf("%s\n", envp[i]);
21     }
22     /* TODO: Find out why using only argv[1] and argv[2] below does not work
23      * with strdup, string is name=value
24      * without strdup, string is name=
25      */
26     envp[i] = strcat(strcat(strdup(argv[1]), "="), strdup(argv[2]));
27     envp[i + 1] = NULL;
28     for (i = 0; envp[i] != NULL; i++) {
29         printf("%s\n", envp[i]);
30     }
31     /* man 7 environ
32      * GNU_SOURCE not defined
33      * cpp -dM /dev/null | grep "_GNU_SOURCE" outputs nothing
34      */
35     extern char **environ;
36     int i, rewrite;
37     if (strcmp(argv[3], "Y")) {
38         rewrite = 1;
39     }
40     else if (strcmp(argv[3], "N")) {
41         rewrite = 0;
42     }
43     else {
44         printf("Usage ./print_env_var <env_var_name> <env_var_value> <rewrite (Y/N)>\n");
45         exit(0);
46     }
47     for (i = 0; environ[i] != NULL; i++) {
48         printf("%s\n", environ[i]);
49     }
50     /* putenv can also be used
51      */
52     if (setenv(argv[1], argv[2], rewrite) == -1) {
53         perror("Setenv failed\n");
54         return errno;
55     }
56     /* oddly enough, in this case too env | grep "name" outputs nothing
57      * if this is the case, the code commented above also works the same way
58      * this one does
59      */
60     /* this executable will execute in a child process whose parent process is
61      * the running shell (bash) and a child inherits the parent's env vars but
62      * can only set its own env vars and cannot modify those of the parent
63      * this follows the concept of copy-on-write between a parent and its child,
64      * the process memory is shared by the parent and the child till the any one
65      * of them tries to make modifications to the content shared by them, at
66      * this point a separate copies are made for the parent and child and
67      * whoever wants to makes modifications can then do so accordingly and these
68      * will not be reflected in the other process (parent or child)
69      * env | grep "name" outputs nothing because it shows the env vars of the
70      * bash i.e. the parent process of our executable's process in this case and
71      * changes in the child's env vars will not be reflected in the parent's env
72      * vars
73     */
74     NORMAL ./print_env_var.c
75     unix < utf-8 < c 47% 37:12
```

```
assignment_4 : vi — Konsole
File Edit View Bookmarks Settings Help
1 ./print_env_var.c
73 //
74 for (i = 0; environ[i] != NULL; i++) {
75     printf("%s\n", environ[i]);
76 }
77 return 0;
78 }
```

## Output-

```
assignment_4 : zsh — Konsole
File Edit View Bookmarks Settings Help
niranay@niranay ~ - /Documents/sem7/AUP/assignment_4 ./print_env_var name value N
XDG_VTNR=7
GDBHISTSIZE=
SSH_AGENT_PID=1657
PAM_KWALLET5_LOGIN=/tmp/kwallet5_niranay.socket
XDG_SESSION_ID=2
TERM=xterm-256color
SHELL=/bin/bash
KONSOLE_DBUS_SERVICE=:1.33
CSAT_PATH=/home/niranay/Documents/disk-backup-master-6079755f1e7ba3fa3447cad429df4cec7da2c6bb//Documents/CSAT/CSAT2/GS-Code/
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
KONSOLE_PROFILE_NAME=Profile #1
GS_LIB=/home/niranay/.fonts
WINDOWID=83886085
SHELL_SESSION_ID=39c7e9a1739e40009b545df5b1ce4e92
KDE_FULL_SESSION=true
XDG_SESSION_CLASS=user
USER=niranay
LS_COLORS=rs=0:di=01;34:ln=01;34:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32*:tar=01;31*:tgz=01;31*:arc=01;31*:arj=01;31*:taz=01;31*:lha=01;31*:lz4=01;31*:lzh=01;31*:lzma=01;31*:tlz=01;31*:txz=01;31*:tzo=01;31*:t7z=01;31*:zip=01;31*:z=01;31*:Z=01;31*:dz=01;31*:gz=01;31*:lrz=01;31*:lz=01;31*:lzo=01;31*:xz=01;31*:bzip=01;31*:bz=01;31*:tbz=01;31*:tbz2=01;31*:tz=01;31*:deb=01;31*:rpm=01;31*:jar=01;31*:war=01;31*:ear=01;31*:sar=01;31*:rar=01;31*:alz=01;31*:ace=01;31*:zoo=01;31*:cpio=01;31*:7z=01;31*:rz=01;31*:cab=01;31*:jpg=01;35*:jpeg=01;35*:gif=01;35*:bmp=01;35*:pbm=01;35*:pgm=01;35*:ppm=01;35*:tga=01;35*:xbm=01;35*:xpm=01;35*:tif=01;35*:tiff=01;35*:png=01;35*:svg=01;35*:svgz=01;35*:mng=01;35*:pcx=01;35*:mov=01;35*:mpg=01;35*:mpeg=01;35*:m2v=01;35*:mkv=01;35*:webm=01;35*:ogm=01;35*:mp4=01;35*:m4v=01;35*:mp4v=01;35*:vob=01;35*:qt=01;35*:nuv=01;35*:wmv=01;35*:asf=01;35*:rm=01;35*:rmvb=01;35*:flc=01;35*:avi=01;35*:fli=01;35*:flv=01;35*:gl=01;35*:dl=01;35*:xcf=01;35*:xwd=01;35*:yuv=01;35*:cgm=01;35*:emf=01;35*:ogv=01;35*:ogx=01;35*:aac=00;36*:au=00;36*:flac=00;36*:m4a=00;36*:mid=00;36*:midi=00;36*:mka=00;36*:mp3=00;36*:mpc=00;36*:ogg=00;36*:ra=00;36*:wav=00;36*:oga=00;36*:opus=00;36*:spx=00;36*:xspf=00;36:
QT_ACCESSIBILITY=1
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session1
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/tmp/ssh-XkbrdIVCln5j/agent.1616
KICAD_PATH=/usr/share/kicad
SESSION_MANAGER=local/niranay:@/tmp/.ICE-unix/1918,unix/niranay:/tmp/.ICE-unix/1918
DEFAULTS_PATH=/usr/share/gconf//usr/share/xdg/sessions/plasma.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg:/usr/share/xdg/sessions/plasma:/etc/xdg:/usr/share/kubuntu-default-settings/kf5-settings
KIDTHUB=https://github.com/KiCad
DESKTOP_SESSION=/usr/share/xdg/sessions/plasma
XDG_PATH=/home/niranay/bin:/home/niranay/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
QT_IM_MODULE=compose
```

```
assignment_4 : zsh — Konsole
File Edit View Bookmarks Settings Help
PATH=/home/niranay/bin:/home/niranay/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
QT_IM_MODULE=compose
PWD=/home/niranay/Documents/sem7/AUP/assignment_4
XDG_SESSION_TYPE=x11
KONSOLE_DBUS_WINDOW=/Windows/1
KDE_SESSION_UID=1000
LANG=en_IN
MANDATORY_PATH=/usr/share/gconf//usr/share/xdg/sessions/plasma.mandatory.path
GCC_COLORS=error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01
KONSOLE_DBUS_SESSION=/Sessions/6
SHLVL=1
COLORFGBG=15;0
HOME=/home/niranay
XDG_SEAT=seat0
KDE_SESSION_VERSION=5
LANGUAGE=en_IN:en
XCURSOR_THEME=breeze_cursors
LOGNAME=niranay
XDG_SESSION_DESKTOP=KDE
CLASSPATH=/usr/share/java/mysql.jar
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-vMBXDh0jx8,guid=36659a198fb993a5e542fac85d70e39e
XDG_DATA_DIRS=/usr/share:/usr/share/xdg/sessions/plasma:/usr/local/share:/usr/share:/var/lib/flatpak/desktop:/var/lib/flatpak/desktop
KDED_STARTED_BY KDEINIT=1
LESSOPEN=| /usr/bin/lesspipe %s
PROFILEHOME=
DISPLAY=:0
XDG_RUNTIME_DIR=/run/user/1000
XDG_CURRENT_DESKTOP=KDE
LESSCLOSE=/usr/bin/lesspipe %s %s
PAM_KWALLET_LOGIN=/tmp/kwallet_niranay.socket
XAUTHORITY=/tmp/xauth-1000-0
OLDPWD=/home/niranay/Documents/sem7/AUP/assignment_4
ZSH=/home/niranay/.oh-my-zsh
PAGER=less
LESS=-R
LS_COLORS=GxFcxdxbxegebabagacacd
_=/home/niranay/Documents/sem7/AUP/assignment_4/./print_env_var
name=value
niranay@niranay ~ - /Documents/sem7/AUP/assignment_4
```